National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

NL-339 (r. 88/04)

Canada

THE UNIVERSITY OF ALBERTA

SIMULATION AND CONTROL OF A ROD MILL-BALL MILL GRINDING CIRCUIT

by

© ANDREW JAMES NEALE

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

IN

MINERAL ENGINEERING

DEPARTMENT OF MINING, METALLURGICAL AND PETROLEUM ENGINEERING

EDMONTON, ALBERTA

FALL 1987

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-41177-5

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR ANDREW JAMES NEALE

TITLE OF THESIS SIMULATION AND CONTROL OF A ROD MILL-BALL MILL GRINDING CIRCUIT

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED FALL 1987

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Andrew Neale

PERMANENT ADDRESS:

1265 ARVO STREET

SUDBURY, ONT.

P3A 3P1

DATED JUNE 26 1987

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled SIMULATION AND CONTROL OF A ROD MILL-BALL MILL GRINDING CIRCUIT submitted by ANDREW JAMES NEALE in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE in MINERAL ENGINEERING.

.....

Supervisor

.....

.....

Date...June...26....1987.....

Dedication

To Espie

Thanks for putting up with me, or without me, as the case may be, it can only get better when we get to Sudbury

and to Michael

Hopefully you will soon realize that the stranger who shows up at the house for a couple of hours every day to eat is actually your father

Abstract

It has long been recognized in the processing industry that a valid dynamic simulator is an indispensible tool in the design and evaluation of automatic control schemes. In the mineral processing field in particular the lack of such a simulator is felt to be a major hindrance to the widespread implementation of effective controllers. The major thrust of this current work has been the development of an interactive rod mill-ball mill grinding circuit dynamic simulator that can be used to develop and test various feed-back control strategies. The software developed provides a powerful educational and research tool for those individuals interested in grinding circuit control applications.

A complete review of the status of dynamic simulation in mineral processing is provided which indicates the need for this type of simulation package. A review of the simulator, highlighting the modular structure and open-ended design, demonstrates the ease with which the basic package can be modified in order to meet individual requirements. Various open-loop simulations are presented to demonstrate the highly interactive nature of the grinding circuit and to indicate where potential control problems may be encountered. Closed-loop simulations are examined, starting from the basic single input single output classical feedback control of the product stream size distribution. Further improvements in the grinding circuit performance in the presence of various disturbances is realized through the application

of a classical multivariable feedback controller. The implementation of static decouplers to reduce the interaction problem was found to be very effective, and resulted in addition improvements in the overall circuit performance. An IBM PC version of the simulator is currently available at cost through the Department of Mining, Metallurgical and Petroleum Engineering, University of Alberta, to any interested individuals.

Acknowledgement

Although only one person is credited with writing a thesis, very few of those can honestly claim to have done so without the help of others. In this case, the author is indebted to various individuals for their assistance in completing this project. In particular, my supervisor Dr. Brian Flintoff is to be thanked for allowing me the freedom to develop my own ideas and make my own mistakes, but always being there when some of the technical problems seemed insurmountable. The assistance of Dr. Reg Wood of the Department of Chemical Engineering with various aspects of the controller design work is greatly appreciated. Mr. Doug Booth also provided invaluable help in the area of software development and preparing the PC version of the simulator. Lastly the author would like to thank the cohabitants of 280E who made life bearable when the project wasn't going well, and unbearable when it was. It could have been done without you, but it wouldn't have been half as much fun.

Table of Contents

Chapter — Page

Dedication iv

Abstract v

Acknowledgement vii

1. Introduction 1

2. Modelling and Simulation in Mineral Processing 3

2.1 Modelling and Simulation 3

2.1.1 Models and Modelling 3

2.1.2 Simulation 6

2.1.3 General Applications of Dynamic Simulation 8

2.2 Simulation in Mineral Processing 14

2.2.1 A Brief History 14

2.2.2 Dynamic Simulation of Unit Operations 19

2.2.2.1 Crushing 20

2.2.2.2 Grinding 23

2.2.2.3 Flotation 25

2.2.3 Dynamic Simulation Software 26

3. Grinding Circuit Dynamic Simulation 30

3.1 Grinding Circuits 34

3.2 Simulator Models 38

3.2.1 Mill Model 39

3.2.2 Hydrocyclone Model 49

3.2.3 Fixed Time Delay 55

3.2.4 Variable Time Delay 57

3.2.5 Sump and Pump Models 60

3.3 Simulator Structure 68

3.3.1 The Executive 69

3.3.2 COMMON BLOCKS 81

3.3.3 DYFLO Utilities 86

3.4 Open Loop Simulations 89

3.4.1 Increase in Fresh Ore Feed Hardness 93

3.4.2 Increase in Sump Water Flow Rate 96

3.4.3 Increase in Fresh Ore Feed Rate 99

3.4.4 Change in Feed Size Distribution 102

4. Classical Feedback Control of the Grinding Circuit Dynamic Simulator 105

4.1 Implementing Control on the Simulator 112

4.2 Design and Evaluation of Feedback Controllers ..114

4.3 Implementing Classical Control on a 2x2 System .118

4.3.1 Controlling Sump Level with Sump Water Flow Rate 122

4.3.2 Controlling Sump Level with a Variable Speed Pump 131

4.4 Summary 142

5. Type I and Type II Control 145

5.1 Interaction Analysis 146

5.2 Implementing Type II Control 152

5.3 Evaluating Type II Control 158

5.4 Servo Control 164

5.5 Type II Control with Static Decouplers 168

5.6 Type II Control with Static Decouplers Using a Fixed Speed Cyclone Feed Pump 178

5.7 Summary 184

6. Conclusions and Recommendations 186

6.1 Conclusions 186

6.2 Recommendations 189

6.3 Simulator Availability 192
7. Bibliography 194
8. APPENDIX A 200

List of Figures

Figure		Page
3.1	Grinding Circuit Configurations Currently Included in the Simulator	35
3.2	Partition Curve as Predicted by the Simulator Hydrocyclone Model for Components of Differing Densities	53
3.3	Simulator Generated Output Comparing the Product Stream Size Distribution of a Soft Heavy Component to that of a Hard Light Component	54
3.4	Circular Buffer: with no movement across cell boundaries	56
3.5	Straight Line Buffer: with movement across cell boundaries	56
3.6	Open Loop Simulation of a 20% Increase in Sump Water Flow Rate at t=2 Minutes using a Single Speed Pump	64
3.7	Correlation Between Pump Curves and Model Predictions for a 10" x 8" SRL Pump	67
3.8	Schematic Diagram of the Open-Loop Simulator Structure	70
3.9	Schematic of Reversed Closed Circuit Simulation Flowsheet	78
3.10	Schematic of Conventional Closed Circuit Simulation Flowsheet	79
3.11	Schematic of Open Circuit Simulation Flowsheet	79
3.12	Open-Loop Response to an Increase in Ore Hardness at t=2 Minutes	94
3.13	Open-Loop Response to a 20% Increase in Sump Water Flow Rate at t=2 Minutes	97
3.14	Open-Loop Response to a 20% Increase in the Fresh Ore Feed Rate at t=2 Minutes	100
3.15	Open-Loop Response to an Change in the Fresh Ore Feed Stream Size Distribution	103
4.1	Process Matrix for a Grinding Circuit (after Herbst and Rajamani)	110

Figure Page

4.2 Linking of Controlled and Manipulated Variables for Classical Control Strategies (after Herbst and Rajamani) 110

4.3 Schematic Diagram of the Closed-Loop Simulator Structure 115

4.4 FOF Controlling COS, Ore Hardness Increase at t=2 Minutes, Fixed Speed Pump, (K_c=11.95, τ_I=4.07) 121

4.5 Open-Loop Response of Grinding Circuit to Changes in Sump Water Flow Rate 124

4.6 Circuit Response to an Increase in Hardness at T=2 Minutes with Sump Water Flow Rate Controlling Sump Level (K_c=1, τ_I=0.25) 125

4.7 FOF controlling COS (K_c=10.5, τ_I=7) and SWF controlling SL (K_c=1), IAE=14.6 129

4.8 FOF controlling COS (K_c=10.5, τ_I=8) and SWF controlling SL (K_c=0.25, τ_I=4), IAE=20.0 130

4.9 Open-Loop Response of Grinding Circuit to Changes in Pump Speed 132

4.10 Control of Sump Level by Manipulation of Pump Speed in Response to a Step Change is Sump Water Flow Rate 136

4.11 FOF controlling COS (K_c=4, τ_I=15) and PS controlling SL (K_c=2,), IAE=113.3 137

4.12 FOF controlling COS (K_c=5, τ_I=10) and PS controlling SL (K_c=2,), IAE=161.3 139

4.13 FOF controlling COS (K_c=14, τ_I=7) and PS controlling SL (K_c=0.5,), IAE=19.5 141

5.1 Open-loop Response of Circuit to a 20% Decrease in Fresh Ore Feed rate 148

5.2 Open-loop Response of Circuit to a 20% Decrease in Sump Water Flow rate 149

5.3 Type II Control, Response to a Step Increase in Ore Hardness at t=2 Minutes 154

5.4 Type I Control, Response to a Step Increase in Ore Hardness at t=2 Minutes 157

Figure Page

5.5 Type II Control, Step Increase in Ore Hardness at t=2 Minutes, Step Decrease in Ore Hardness at t=202 Minutes 159

5.6 Type II Control, Extraneous Water Flow to Sump - 9.6 tph for 100 minutes, starting at t=2 minutes 161

5.7 Type II Control, Step Increase in Rod Mill Feed Coarseness at t=2 minutes 163

5.8 Servo Control, Controller Response to COS Set-Point Changes 165

5.9 Servo Control, Controller Response to CL Set-Point Changes 167

5.10 Block Diagram Representation of the Closed-Loop Grinding Circuit 169

5.11 Block Diagram of Grinding Circuit with Static Decouplers Included 172

5.12 Open-Loop Response to a 10 tph Increase in SWF using Static Decouplers 175

5.13 Open-Loop Response to a 10 tph Decrease in FOF using Static Decouplers 176

5.14 Type II Control with Static Decouplers, Response to a Step Increase in Ore Hardness at t=2 Minutes 177

5.15 Type II Control with Static Decouplers, Response to a Step Increase in Ore Hardness at t=2 Minutes, Step Decrease in Ore Hardness at t=202 Minutes 179

5.16 Type II Control, Static Decouplers, Fixed Speed Pump, Response to a Step Increase in Ore Hardness at t=2 Minutes 181

5.17 Type II Control, Static Decouplers, Fixed Speed Pump, Response to a Step Increase in Ore Hardness at t=2 Minutes, Step Decrease at t=202 Minutes 183

1. Introduction

The main emphasis of this work has been to develop a valid rod mill-ball mill dynamic simulator that will provide a powerful tool for the design and evaluation of grinding circuit control systems, to be used in both the university and industrial environment. This work is presented in five major chapters, which basically follow the progress of the project from its inception through to the final analysis of the simulator performance.

The first chapter documents a detailed literature survey which reveals the need for a valid, dynamic simulator for an industrial scale rod mill-ball mill grinding circuit. A review of the dynamic models currently available lays the groundwork for the development of the simulator. The subsequent chapter provides a detailed description of the models used in the simulator. In addition, the structure of the simulator is discussed emphasizing the modularity and open-endedness of the package. Since future user expansion is encouraged, a complete discussion of the simulator is necessary to prepare the user for running and modifying the simulator. The chapter concludes with the presentation of various open-loop simulations which validate the simulator code and indicate the interaction problems that will have to be dealt with in order to implement control.

The control problem is introduced in Chapter 4 with a brief review of classical feedback control theory. A study of the control of the product stream size distribution by

manipulation of fresh ore feed rate, and adjusting either pump speed or sump water flow rate to control sump level highlights some of the expected difficulties. Significant detuning of one or both of the control loops is required in order to implement satisfactory control. The use of a variable speed pump to control sump level is justified so that Type I and Type II control strategies can be evaluated in Chapter 5.

Interaction analysis indicates that the Type II controller provides the appropriate pairing of inputs and outputs. Simulations validate this hypothesis, and good control is achieved by using sump water flow rate to control the cyclone overflow size distribution and fresh ore feed rate to control circulating load. Various disturbances are examined in order to study the capabilities of the controller. Further improvement in performance is gained by implementing static decouplers to reduce the effects of loop interactions. The chapter concludes with the indication that the controller is just as effective when a fixed speed pump is used, and sump level is allowed to fluctuate.

The final chapter provides a review of the major results, and includes recommendations for possible future work.

2. Modelling and Simulation in Mineral Processing

2.1 Modelling and Simulation

Mathematical modelling and simulation is well recognized today as a fairly inexpensive way of yielding insight into the behaviour of an industrial process. Simulation has been shown to be faster, less expensive and more accurate than traditional experimental techniques for acquiring an understanding of a particular system and its behaviour under various operating conditions. Although it would appear in the literature as though many workers use the terms modelling and simulation synonymously, this author would rather keep them separate. Modelling, it is felt, should refer to the development of the physical or mathematical models of a system, whereas simulation should refer to the application of these models in order to gain knowledge of the system behaviour. This philosophy is echoed by other workers, e.g. Lynch (1977) who narrowly defines simulation as the representation of a physical process by mathematical models. This differentiation may seem trivial at first, but it does allow for further categorization of the modelling and simulation process.

2.1.1 Models and Modelling

Models can be defined as the building blocks used to describe individual units within a system, and can take various forms. The first type are known as iconic models

which are simply scaled, physical objects. These have found extensive use in the design departments of the aeronautical and automotive industries. In the processing industry, scale models have found some limited use in the plant design and architectural studies, but more often than not are simply used for demonstration. A second group known as visual models use graphical representation of a system and are usually designed for non-technical presentation of information. That is to say, the system output to a given set of inputs is defined without any necessary understanding of the actual system. Such applications are commonly found in the areas of economics and business administration. The last category of models, and the one of primary importance to process engineers, is abstract models, in which mathematical equations and relationships are used to describe a system. It is this category that will be dealt with in all further discussions.

Mathematical modelling is a complex and diverse field, and as such the models used can be further sub-divided into three general types. Empirical models are derived from experimental observation around the system in question. They do not explicitly account for the process fundamentals; instead they use standard regression techniques to develop a linear or non-linear correlation between the system outputs and inputs. Empirical modelling is the simplest modelling method, often known as the 'black box' approach, but has applications limited to the experimental region for which the models are derived. Extrapolation of empirical models has

certain inherent dangers and should be avoided. A more complex approach is known as phenomenological modelling in which theory provides the general form of the model, and the model parameters are developed from experimental observation. In this way conservation laws for mass, momentum and energy can be used to develop accurate models without the need for interpretation or extraplolation of the underlying theory. The third general class of models are the mechanistic models, also known as fundamental ~~or~~ theoretical models, which utilize the basic laws of chemistry and physics. Although the most accurate, mechanistic models require a complete understanding of the underlying theory of a given system such that all parameters are known *a priori*. The mineral processing engineer has little use for this last class of models due to the heterogeneous nature of the material involved and, in the case of the simulation of comminution circuits, the lack or a complete understanding of the processes involved. However, the former two classes of models, empirical and phenomenological, have found extensive use in the mathematical modelling of unit operations in mineral processing. All of the models utilized in this work fall into these two categories, with the former describing the hydrocyclone classifier performance and pump head capacity curves, while the latter describes the actual comminution process.

2.1.2 Simulation

Since this work is concerned in part with dynamic simulation in mineral processing, it is beneficial at this point to briefly cover some of the nomenclature involved. Simulation of any system can take many forms, varying from the physical use of iconic models to the more abstract, involving the implementation of a set of mathematical models to describe a given system. The former is of little importance in the area of process design, whereas the latter has received a great deal of attention from the process engineer. Due to the nature of mathematical simulation, with its iterative use of algebraic and/or differential equations, it is implemented almost exclusively on digital computers. As a result the terms mathematical simulation and computer simulation are considered synonymous and are often used interchangeably in the literature. Bekey (1975) has gone as far as defining simulation as the implementation of mathematical models of physical systems on digital computers; a definition that would certainly seem appropriate to this work.

Like modelling, it is often useful to categorize simulation. The two general types of simulation being deterministic and stochastic. The former indicates that for a given set of inputs a particular set of outputs will be observed with these outputs being observed every time the simulation is performed. In contrast, a stochastic simulation will yield varying sets of outputs for a given set of inputs each time the simulation is run, given that the seed of the

pseudorandom number generator is altered for each simulation (Pritsker, 1984) Such simulations are performed in order to model some form of random behaviour within the system that cannot be measured or predicted, but has a known probability-distribution. Both types of simulation can include steady state or dynamic simulation.

Steady state simulators, as the name implies, enable the prediction of the steady state behaviour of a given system. This usually involves the iterative solution of a set of algebraic equations. The steady state solution is reached once the most recent iteration is within a certain tolerance of the previous solution set. These steady state simulators enable prediction of the performance of an individual unit through to the entire plant flowsheet, for a given set of design or operating conditions. Steady state simulators can also include economic models to assist in decisions on plant modernization, etc.

Dynamic simulators, on the other hand, can be applied to any system in which the transient behaviour is of interest. This could involve, for example, moving the system from one steady state to another, or monitoring or controlling the effects of time varying inputs and disturbances into the sytem. The common thread within dynamic simulation is that time is used as the independent variable and the models are described by ordinary or partial, linear or non-linear differential equations. Dynamic simulators for control purposes can often utilize simplified, linearized models since, in

general, they don't need to operate over a wide range. However, in order to perform the simulations in the areas of system start-up and shut-down, fairly complete dynamic models are required, with the concurrent increase in modelling effort and computing requirements. Since this current work involves the evaluation of controller configurations on grinding circuit performance, it is assumed that the operating ranges within the system will be fairly narrow. As such, the simulator is not currently designed to model start-up and shut-down procedures, a point the reader is asked to keep in mind while covering this material. For those users interested in studying such applications, it may be possible to modify the simulator by including the appropriate constraints and decision blocks.

2.1.3 General Applications of Dynamic Simulation

Due to the highly non-linear nature of the mathematical models encountered in mineral processing, and the difficulties in developing reliable on-line sensing instruments for implementing control, the volume of material published by the minerals industry in regards to dynamic simulation, although significant, is dwarfed by that produced from the nuclear, chemical and U.S. defence industries. There is of course an advantage in being the late player in the field in that it is possible to learn from the mistakes of others, utilizing what is felt to be useful and applicable in the minerals industry and, in effect, shortening the learning

curve for applying dynamic simulation to mineral processing.

The most common application of dynamic simulation in the minerals industry (or any process industry for that matter) is the off-line evaluation of process control strategies. Initially, in the study of classical control theory, this involved the analysis of single input single output (SISO) systems, simulating the open loop response and deriving the PID controller constants. Closed loop simulations could then be performed in order to tune the controllers to provide the best estimates of the plant controller constants. Of historical importance in this regard is the work of Stewart (1970) in which various feedback control schemes were studied and evaluated through computer simulation. Similarly, simulators could be used to determine the best feedforward and/or cascade control systems (Govind, et al., 1982). These investigations worked well in certain applications, but tended to have limitations in many industrial situations such as overall plant control. Although it is possible to isolate SISO systems within a concentrator, there are problems encountered in implementing classical control due to system interactions. As will be discussed in more detail later, scenarios can be realized in which manipulation of one of the inputs affects the performance of more than one of the outputs resulting in detuning of one or more of the loops in order to achieve satisfactory control. Recently there has been a lot of interest in applying what is known as modern control theory to mineral processing

circuits. This has also been referred to as a model-based control strategy since the mathematical models of the system in question are an integral part of the control process, (Rajamani 1979). Such work is beyond the scope of this current project, but does provide for some interesting comparisons with the classical control schemes as discussed by this author. Although fully operational pilot plant facilities for testing control strategies have been constructed, they are extremely expensive and of limited use to the onsite engineer. Simulation on the other hand can be performed on a desk-top computer at a fraction of the cost.

A second major use of dynamic simulators is for operator training. Flight and navigation simulators are perhaps the most sophisticated dynamic simulators in use today. The capital intensive nature of these industries, and life threatening nature of system failures, justifies the expense of developing the simulation instrumentation and hardware. For the control of nuclear power plants, with the obvious inherent dangers of system failure, the use of training simulators is well recognized. To date there hasn't been the same impetus for such sophisticated operator training systems in the minerals industry. One could certainly hypothesize though that there will be a demand for mineral processing operator training simulators in the near future. With the increased reliability of process instrumentation, the frequency of operational deviations and plant stops and starts can be expected to diminish. As a result operators

may tend to lose the *hands on feel* of the process, and have difficulty maintaining their skills for the times when manual control of a process is required. With the use of a training simulator operators could practice handling unusual situations without upsetting the real process. de Jong and de Wijn (1984) give a brief review of the role of simulators for process operator training and applications in the chemical industry. As indicated by Alliston et al. (1971) the requirements for training simulators are far greater than those for design simulators. The emphasis of operator training systems is on visual displays and man-machine interaction. The mathematical models have to operate in real time and be sufficient in detail and accuracy to fully duplicate the true process in the eyes of the operator. In addition, process control room mock-ups, complete with the appropriate instruments and control switches, etc., may be constructed and interfaced with the simulator. It usually takes a team of specialists ranging from the processing engineer to the computer hardware and software experts to construct such training simulators. As a result, these simulators are very expensive and usually unique to a given operation with little or no transportability. These factors combine to make such systems quite unattractive to the minerals industry at present.

At the university level dynamic simulators are slowly being put into use for the training of processing engineers. Although not fully interactive, such as flight simulators,

these software packages are becoming a useful tool for undergraduate and graduate training. Obviously such simulators are not intended to train students to become plant operators, but are required in order for the student to gain a better understanding of the circuit dynamics. A dynamic simulator aids the user in identifying cause and effect relationships within the circuit and selecting the appropriate variables for control strategies. In addition, the nature of the response to any particular disturbance can be ascertained. All this can be done in the university environment, using little more than a desk top computer and some form of hard copy printer. This then exposes the student to various scenarios within the circuit, scenarios which occur rarely in an operating plant. For example, few if any mill superintendents would be willing to overload a grinding circuit to the point of causing a rod tangle, simply in order for a student to observe the effects on the rest of the circuit. Such conditions can be programmed into the simulator with the appropriate warnings being flashed to the user via the CRT should unusual or emergency situations occur. These type of equipment and operating constraints have been included in the crushing plant dynamic simulator developed by Tucker and Meech, 1985, as will be discussed in a subsequent section. Once a suitable dynamic simulator is built, it can be improved upon and enlarged as the user requirements increase. Given a well designed executive structure it is not that difficult to change the unit module subroutines, improve the

user friendliness of the front end, alter the nature and type of data output, and test various circuit and controller configurations, with the only constraints being the ambitions and requirements of the user groups.

2.2 Simulation in Mineral Processing

2.2.1 A Brief History

As indicated by Lynch (1985) in a review of the first twenty-five years of the use of computers in mineral processing, digital computers have been in the industry since the late 1950's. However, as late as 1969 it was noted that complete optimization of mineral processing was impossible due to the lack of appropriate sensing and controlling devices (Spedden, et al. 1970). In addition, lack of the mathematical models needed to describe full scale unit operations in detail hindered further process improvements. Most plants constructed in the late '60's included fully integrated process monitoring systems and automatic control capabilities over such variables as ore feed rate, pulp densities, sump and bin levels, etc. The improved plant performance realized with such systems demonstrated the need for further work in this area. At this time (ie. the late '60's) the digital computer was being recognized for its capabilities in the areas of metallurgical calculations. As a result, the concept of using digital computers for material balance calculations, steady state simulations, flowsheet design, various levels of digital control and eventually dynamic simulation can be traced back to the mid to late 1960's. The increased availability of suitable desk top computers and the development of relatively easy to use high level languages aided this development.

By the early 1980's, so much work had been documented in this area that review papers were published in order to keep workers appraised of the situation. Ulsoy and Sastry (1981) included over 150 references in a review of the concepts, terminology and practice of mineral process control. Interestingly, over half of the papers referenced dealt specifically with the modelling and simulation facets of the control implementation process. By this time, improvements in instrumentation, including the development of reliable particle size analyzers and chemical composition analyzers, greatly enhanced the monitoring of system performance. Concurrent development of dependable final control elements further aided the implementation of computer control schemes. Although the beneficial aspects of automatic control of mineral processes had been documented, (eg. Perkins, et al., 1978; Manning, 1978), Ulsoy and Sastry (1981) indicated at this point that the application of control theory to mineral processing lagged far behind that of the chemical processing industry.

The development of mathematical models of grinding mills had matured to the stage that reliable grinding simulators could be produced. The development of flotation models lagged behind slightly due to the highly complex nature of the flotation process. Several steady state crusher models were also in use at this time. It is important to note though that workers had clearly recognized that an improvement in the performance of the grinding circuit could

potentially reap the greatest economic rewards. This can be attributed primarily to the energy intensive nature of grinding and the concurrent inefficiency of the power input in actually reducing particle size. It was also well recognized that the grinding product strongly affects downstream processes, thus by improving the grind, improvements in the subsequent processes could be realized. Fortunately, the relatively simple, well developed grinding models available at this time facilitated off line control system analysis and design. Although not of primary importance in this work, it is only fair to indicate that simulation and control of crushing and flotation circuits was well under way by the early 1980's, and producing results that were as promising as those experienced in the control of grinding circuits (eg. Bolles, et al., 1977). Of course, although these were not well documented, most researchers will admit that there were some dismal failures in the attempts to implement control strategies. The author can only surmise that these failures were due in part to an incomplete understanding of the process to be controlled, the lack of appropriate hardware and software, or perhaps unrealistic expectations of what any given control system could do for an operation.

The decade of the eighties has seen further gains in developing an understanding of modern control theory and its potential in mineral processing. It is probably fair to say that the leading researchers in this field today are on par with any of their colleagues in other process control fields

such as petrochemical plants, nuclear power stations and the manufacturing industries. Unfortunately a point has now been reached in which the knowledge gained in research is not filtering down to industry fast enough. As Lynch (1985) indicates, even today there is still a reluctance to use computers in the plant for anything more sophisticated than data logging and simple control duties. This in spite of the fact that Weiss (1985) has shown that the application of computers to process control yields a benefit/cost ratio of up to 8:1, where this ratio is defined as the ratio of the annual savings to the sum of the annual operating cost of the computer system plus one year's amortization of the development cost. This ranks process control in the top three most profitable applications of computers in the minerals industry, along with maintenance and inventory control and mine planning.

Perhaps one of the major reasons for the resistance to implementing complex computer control systems in mineral processing plants is the enormous amount of work and expertise required. This of course is in addition to the daily duties required of the plant engineer. Due to the unique nature of almost any given ore, and for that matter a lot of the coals, each concentrator is individually designed, and as a result each control scheme has to be individually designed. This design work has to be done without creating major upsets in the circuit that invariably result in some form of direct or indirect financial losses. Such

constraints virtually preclude the use of most open loop tests and factorial design experiments needed to identify the significant controller and process model parameters. The work of Brittan, et al. (1975) gives an indication of the requirements for modelling an entire concentrator, an initial step in any process control study. Although this was an exercise in empirical modelling, and not exactly the same testing and sampling campaign as would be undertaken in control studies, it does provides a useful illustration. Thirty five plant variables were monitored daily over a six month period with multiple regression techniques being used to identify the six significant operating parameters. Such work requires a great deal of engineering man hours and the appropriate computing facilities. An alternative to this procedure is to dynamically simulate the process on a computer, followed by verification of the simulator output with plant data. At this stage experimental control strategies can be tested off line using the simulator. With some tuning to be expected at the implementation stage the best simulated control algorithm can then be used on the real process. The word best is used cautiously since there is still some controversy as to how one exactly defines the optimum performance of a controller, Stephanopoulos (1984). Of course, one could argue that the work required to develop a full scale dynamic simulator of an entire plant would equal or exceed that performed by Brittan, et al.. However, if a general purpose dynamic simulator existed that modelled individual

unit operations, tied together by the appropriate executive structure, then the work required would involve only the parameter estimation for the unit models and some fine tuning of the simulator. This is still a non-trivial problem, but is certainly more expedient than the plant engineer attempting the whole procedure on his own. Such a dynamic simulator does not as yet exist, but various workers are approaching the stage of producing the equivalent general steady state simulators (eg. Ford, et al., 1984; Laguitton, et al., 1983) for flowsheet design and economic feasibility studies. One could surmise though that general plant dynamic simulators will be available soon, certainly within the next five years, if not sooner.

2.2.2 Dynamic Simulation of Unit Operations

Within a mineral processing circuit each piece of equipment can be modelled as an individual unit, with this unit being connected to the rest of the circuit by means of pipes, chutes and conveyors. Herbst and Mular (1979) give a good introductory review of the process of modelling and simulation of unit operations. At present there are suitable dynamic and steady state models available for virtually all the major units in a concentrator, as indicated by Sastry et al. (1984). These include crushers, screens, grinding mills, classifiers, flotation cells, dewatering and thickening units, etc. Niemi (1983) reviews some of the dynamic models currently in use, their steady state counterparts, and some

of the uses and applications of these models. Models have also been developed for circuit holdups such as sumps and bins, pump models have been developed using manufacturer's head capacity curves, and piping and conveyors have been modelled as variable and fixed time delays respectively. Thus all the tools are available to build dynamic simulators for entire plants, but as indicated earlier, this work has not as yet been completed. However, within a given plant three main areas, crushing, grinding and flotation, have been modelled as entire units by various workers. Other unit operations such as dewatering, thickening and clarification have all been studied to a lesser extent, but are beyond the scope of this current work (see e.g. Kelly and Spottiswood, 1982). It is beneficial at this point to briefly review some of the work performed in these three main areas. This will cover the development of the initial steady state models and simulators to the current trends in dynamic simulations. A more detailed discussion of the specific models used in the grinding simulator developed in this work will be given in a subsequent chapter.

2.2.2.1 Crushing

Some of the earliest successful work in modelling crushing plants was performed by Whiten (1972) studying cone crushers. The crusher model was developed by assuming a single breakage zone within the crusher in which particles could either be broken or pass through the crusher unbroken. This simplified the model to a single

breakage zone and a calculated probability of particles entering or re-entering the breakage zone. Thus only a breakage and classification matrix had to be determined experimentally using non-linear least-squares methods. These models were used in steady state simulations of entire crushing plants and were found to be an inexpensive, fast and accurate method of studying various operating variables such as primary and secondary crusher gap, primary and secondary screen apertures sizes and fresh ore feed rate. This model formed the basis for other steady state simulation studies, the most notable of these being the work of Hatch et al. (1982). These workers found the structure of the empirical crusher model of Whiten to be quite satisfactory, and only proposed changes in the nature of the screen models in order to simulate a secondary crushing plant. These steady state simulators proved valuable in off-line optimization and circuit design studies. Whiten (1984) has suggested that dynamic simulations of crushing plants is possible with these models by assuming that they and the screen models act as instantaneous units with a short pure time delay added to the products. This may be satisfactory in some situations, but reveals little information as to the almost immediate consequences of such occurrences as a surge of hard ore into the crushing bowl on the crusher current, etc.

Tucker and Meech (1985) have developed an interactive dynamic simulator specifically designed for use in

a university teaching environment. An unusual approach was taken in modelling the circuit in that

> "design considerations, performance specifications and suggested operating and control procedures of the major equipment manufacturers"

were used to derive empirical models of the system. As a result stream size distribution calculations are not made, instead general directional trends in the circuit are produced in response to changes in operating strategy and ore characteristics. Although it has been proposed for use in developing automatic control strategies, this author will reserve judgement on the simulator until it has been verified against actual operating plant data. Intuitively, one wonders about the usefulness of producing control schemes on a simulator if there is no proof that it and the process being simulated have much in common. However, to date this package has been shown to be useful in the study of the application of fuzzy logic towards controlling mineral processing operations, (Harris and Meech, 1985).

In comparison, Herbst and Oblad (1985) are developing a phenomenological dynamic model of a cone crusher. This model subdivides the crusher bowl and crushing chamber into N well mixed, discrete zones and uses ore dependent breakage functions and machine dependent selection functions to describe the change in particle size within each zone. In addition the transport

characteristics of the ore are modelled as a function of the crusher internal geometry. The proposed model has been verified using a 0.5 tph pilot scale cone crusher. This model in turn forms the basis of DYNACRUSH, a dynamic simulator to evaluate the feasibility of crusher modelling, estimation and control. Although still early in its development, the authors indicate a desire to test this model in industrial situations as a further step in developing control systems for crushing plants. This in turn may eventually lead to a general purpose crushing plant dynamic simulator.

2.2.2.2 Grinding

The number of papers published in the area of modelling and simulation of grinding mills and circuits is probably in the hundreds. Austin (1971) gives a good review of some of the earlier work in the development of the mathematical models that describe grinding. Most workers have considered grinding as a dynamic rate process, and comminution is usually described partially as a function of the amount of time that the material remains in the grinding mill. Austin et al. (1984a, 1984b) have contributed a great deal to the understanding of the comminution process. In addition, the book by Lynch (1977) appears to have become a standard reference for any work in the area of crushing and grinding. Since virtually all grinding circuits today operate in closed circuit with hydrocyclones, a great deal of effort has gone into

developing suitable models of these units. Three researchers, Plitt (1971, 1976) and Lynch and Rao (1975) have introduced empirical models for predicting classifier performance as a function of the operating conditions and hydrocyclone geometry. Very few papers are produced on this subject today without referencing one or all of these articles. The choice of which set or combination of models to use in a given application is often unclear. There appears to be little concensus as to which set of models is better over all operating conditions, both have proved to be satisfactory in various documented simulation studies. Due to the small holdup volume of cyclones, the dynamics can be ignored, and the units are modelled as instantaneous processes using algebraic equations.

The goal of a grinding circuit simulator is to predict the size distribution and flow rates of product streams as a function of the feed ore and circuit operating conditions. Finlayson and Hulbert (1980) have taken this one step further by proposing a liberation model so that the size distribution of the valuable mineral in the product can be described. A circuit simulator was developed using this model in order to test various control schemes. Although it appears that this simulator has been used extensively in South Africa by the Mintek group (formerly NIM), it doesn't as yet appear to be available to the general public. Smith and Guerin (1980) developed a specific dynamic simulator of a Canadian grinding

circuit in order to develop circuit controllers, yet released very little information regarding the simulator. Rajamani (1979) developed a pilot plant grinding simulator, DYNAMILL, in an effort to test various modern control algorithms. This work appears to be very succesful and provides some of the impetus for the research undertaken by this author. Perhaps the most forthcoming work in this area is that of Flintoff et al. (1985) who developed a full grinding circuit dynamic simulator based on the published data of Smith and Guerin (1980). This work in turn forms the basis for the simulator developed by this author the structure of which will be discussed more fully in the next chapter.

2.2.2.3 Flotation

As indicated previously, flotation models are more complex due to the nature of the process. By the mid 1970's King (1975) documented the development of a flotation plant steady state simulator that produced valid results in both pilot plant and full scale operations. Later work in developing dynamic simulators is usually associated with developing appropriate control schemes at individual operations. Two recent papers of note are those of Atkins et al. (1986) and Bascur (1982).

Atkins identified eleven possible control objectives and 15 potential manipulated variables for the control of a chalcopyrite flotation circuit. A largely empirical state space model was found to adequately simulate the

dynamics of the flotation circuit. Model reduction techniques had to be invoked since the computational costs of simultaneously solving the 300 stiff differential equations were prohibitive. However, a sufficiently accurate model was derived in order to tune the desired SISO feedback control schemes using P and PI controllers. Although little information was given as to the nature of the simulator one would be inclined to assume that the simulator was quite specific to the application at hand.

Bascur (1982) utilizes a phenomenological model to simulate the dynamic behaviour of a single flotation cell. The model framework was designed to be very general in nature so that virtually any flotation behaviour could be simulated by altering the sub-processes for a particular ore. A general dynamic simulator, DYNAFLOAT, has been developed in order to aid in the parameter estimation and verification of individual models and as a tool for studying various control stategies.

2.2.3 Dynamic Simulation Software

Since the early 1970's the development of dynamic plant simulators for the chemical process engineer has progressed to such an extent that a substantial number of general purpose dynamic simulators are available to the public (Heyt, et al., 1982). These simulators fall into two major groups, those that use a continuous system simulation language (CSSL), and those that use a high level procedural language

based system, with the language of choice usually being FORTRAN. The former group enjoys a certain degree of standardization (The Simulations Council Inc., 1967), which aids the user, who is faced with a dynamic simulation problem described by a set of differential and algebraic equations, in obtaining a solution with a minimum of programming effort. The latter group tends to be more diverse in nature, and are often quite specific to the original application for which they were intended. Unfortunately, as noted by Wood et al. (1982), it is apparent that most CSSL's are not particularly well suited to handling the multicomponent streams encountered in mineral processing. Another disadvantage of many of the CSSL software systems is that they tend to be fairly expensive, either on a purchase or time share basis, and require a FORTRAN compiler as a separate software item. By developing a FORTRAN based simulator, or any other compiler based high level language, the user requires only the standard compiler. This in turn results in the simulator transportability being very high. As indicated below, interpreter based simulators are being developed which can obviously run on any machine using a common operating system.

Most dynamic simulation efforts in the mineral industry have apparently ignored the work done in the chemical engineering field. Most workers seem to have proceeded in a very independent manner in designing simulators for specific processes. Richardson (1983) has recently reviewed the

development of process simulation in mineral processing. Once again the lack of a general, multi-purpose dynamic simulator is indicated. One of the first efforts at designing and constructing a dynamic simulator of grinding circuits for general use by operating personnel was SIMUMILL (Winslow and White, 1974). Although the authors, then at the University of Arizona, recognized the need and realized the potential use of such software, this system never proceeded much beyond the stage of being a non-specific material balance package. Although various institutions around the world, ie. Mintek (formerly NIM) in South Africa, the Julius Kruttschnitt Mineral Research Centre in Australia and the Technical Research Centre of Finland have developed simulation packages for in-house research and consulting, a certain reluctance to release such work into the public domain is apparent. The Pennsylvania State University group, i.e. Austin et al., have done a great deal of valuable work in grinding circuit simulation, but again don't appear to be interested in producing a general purpose simulation package *per se*. More recently, Adel and Sastry (1982) have reported on the development of a general purpose simulator at the University of California, Berkeley. Named UCMINPRO, this package appears to have some potential, but once again one will have to reserve judgement until the simulator can be validated. Bascur and Herbst (1985) have announced the development of DYNAMILL II and DYNAFLOAT II, both written in BASIC - an interpreter based language - for the simulation of grinding

circuit and flotation dynamics and control. Although BASIC programs tend to execute slower than compiled codes, the main advantage of using BASIC in this application is the ease with which screen graphics can be generated. Obviously computer generated graphics deliver information in a more comprehensive form than the tabulated output expected from FORTRAN programs. These packages are perhaps the first of their kind in that they are designed specifically to train operating personnel, students and plant engineers as to the benefits and design of model-based control strategies. This software is the result of ongoing research at the University of Utah and is part of what appears to be an overall program to develop an entire plant simulator and control scheme.

3. Grinding Circuit Dynamic Simulation

As indicated in the previous chapter, grinding is inherently inefficient and can account for upwards of 50% of the total milling costs. As such, it is an area that is well worth studying in order to improve on the overall performance of the milling operation. Experience in other processing fields, such as petrochemicals and manufacturing, has shown that computer simulation is a powerful tool in gaining an understanding of a particular process, and with this understanding comes the ability to derive and apply methods for improving the process performance. Previous work by this author has indicated that with some effort a steady state grinding circuit simulator can be developed that can be used as an effective tool in evaluating various circuit configurations, feed stream conditions, etc. Such simulators, however, have severe limitations, especially when the transient performance of a circuit is of interest, such as in the case of control studies.

To date, it appears that an effective, general purpose rod mill-ball mill grinding circuit dynamic simulator required for control studies is not readily available. Recently Bascur and Herbst (1985) have developed an interactive dynamic simulator called DYNAMILL II for a single stage, pilot plant scale grinding/classification circuit. Although shown to be effective as an educational tool in designing control strategies, the author feels that it may have severe restrictions in industrial applications. The bulk of this

current chapter is devoted to a discussion of the work performed at the University of Alberta in developing a dynamic simulator which, it is hoped, will meet the needs of those individuals interested in studying the dynamics of industrial scale grinding circuits. This simulator will be a valuable tool in operational, educational and research areas, providing the power and flexibility to simulate various circuit configurations, with or without some form of user specified control algorithm.

The dynamic simulator discussed here is the result of the continuation of work initiated by Wood and Flintoff (1982) at the University of Alberta, and expanded by McDougall (1987). The original simulator was written in FORTRAN IV, whereas the current simulator has been rewritten in order to utilize the FORTRAN 77 compiler and take advantage of the improvements offered by the more efficient coding structure and other facilities not available with FORTRAN IV. Currently there are two versions of the simulator, one that operates on the University of Alberta mainframe computer using the MTS operating system and the FORTRANVS compiler, the second version runs on the IBM PC using DOS and the MICROSOFT FORTRAN compiler.

The major advance in this work has been in expanding the simulator to handle a two-component feed stream. This has resulted in a significant increase in the coding requirements which in turn is reflected in the increased execution time of each simulation. However, the use of a two

component feed stream enhances the simulation of ore hardness and/or density changes, with the former being recognized as one of the major disturbances to grinding circuit operations. For the purposes of this work, the term ore hardness is used to indicate the ore grindability, although the former term tends to be fairly vague, it is used in order to remain consistent with other workers in this area. The software package DYNAMILL II (Bascur and Herbst, 1985) doesn't appear to include hardness changes as a disturbance option, although Rajamani (1979), as part of his doctoral work, used a two component system in developing DYNAMILL, a less interactive forerunner of DYNAMILL II. Further embellishments to the University of Alberta simulator include a flexible front end that allows the user to specify the data input and run time parameters etc., with the final phase of the simulator being directed by the user in order to dump the output to various files for plotting or further data analysis. Finally, various unit modules have been expanded and improved upon, the most notable being the incorporation of a variable speed pump model to replace the single speed pump model used in the previous simulator. This facilitates the control of the sump level in the presence of other circuit disturbances, and its inclusion is justified by the growing acceptance of variable speed pumps in industry.

The simulator performance, throughout its entire development, has been monitored by comparison with the published work of Smith and Guerin (1980). Unfortunately there is to

date a lack of any other appropriate published data on grinding circuit performance. The information required includes the unit model parameters, the circuit stream data such as size distributions, densities, etc., feed and product stream properties and finally, some information regarding the dynamics of the circuit. As a result, the bulk of the simulator is geared to the particular circuit outlined by these authors. It is hoped that in the future, given the existence of a valid dynamic grinding circuit simulator, other workers will be motivated to perform the work required in order to produce the information needed by the simulator. Not until such work is done will the true strengths or weaknesses of the simulator be exposed.

This chapter will put an emphasis on the development of the unit models used in the simulation, identifying the options chosen, and indicating the inherent limitations associated with each of the various models. This first section provides a quantitative look at the modelling requirements of the simulator and presents some of the background material required for a full comprehension of the program. Some thought will be given to possible future expansion and improvements, which given the open structure of the package, can be easily incorporated by individual users as demands dictate. A brief look at the structure of the executive program will give the reader a further insight into how the simulator operates. It is hoped that individuals will be able to identify areas in which this simulator can be used for

particular applications, as well as think of alterations and improvements that will make it more suited to their requirements. The remainder of the chapter will include the results of various open loop simulations, showing the circuit response to different process disturbances. This will verify the performance of the simulator and in turn set the groundwork for control studies to be discussed in the subsequent chapters.

3.1 Grinding Circuits

The types of tumbling or grinding mills, and their configurations in individual flowsheets, are as varied as the types of ores that they are used to process, (Rau and Craig, 1982). It is not the intention of this work to discuss either the operation or layouts of all the possible grinding circuit configurations. The interested reader who lacks in the required mineral processing background is referred to any of introductory texts in this area, (e.g. Wills, 1985). The simulator will perform open-loop simulations of the following grinding circuit configurations, (see Figure 3.1).

1. Open Circuit Rod Mill Ball Mill Grinding
2. Closed Circuit Rod Mill Ball Mill Grinding
3. Reversed Closed Circuit Rod Mill Ball Mill Grinding

The latter, by virtue of its greater capacity, is the most common configuration in use today. As such it will be the

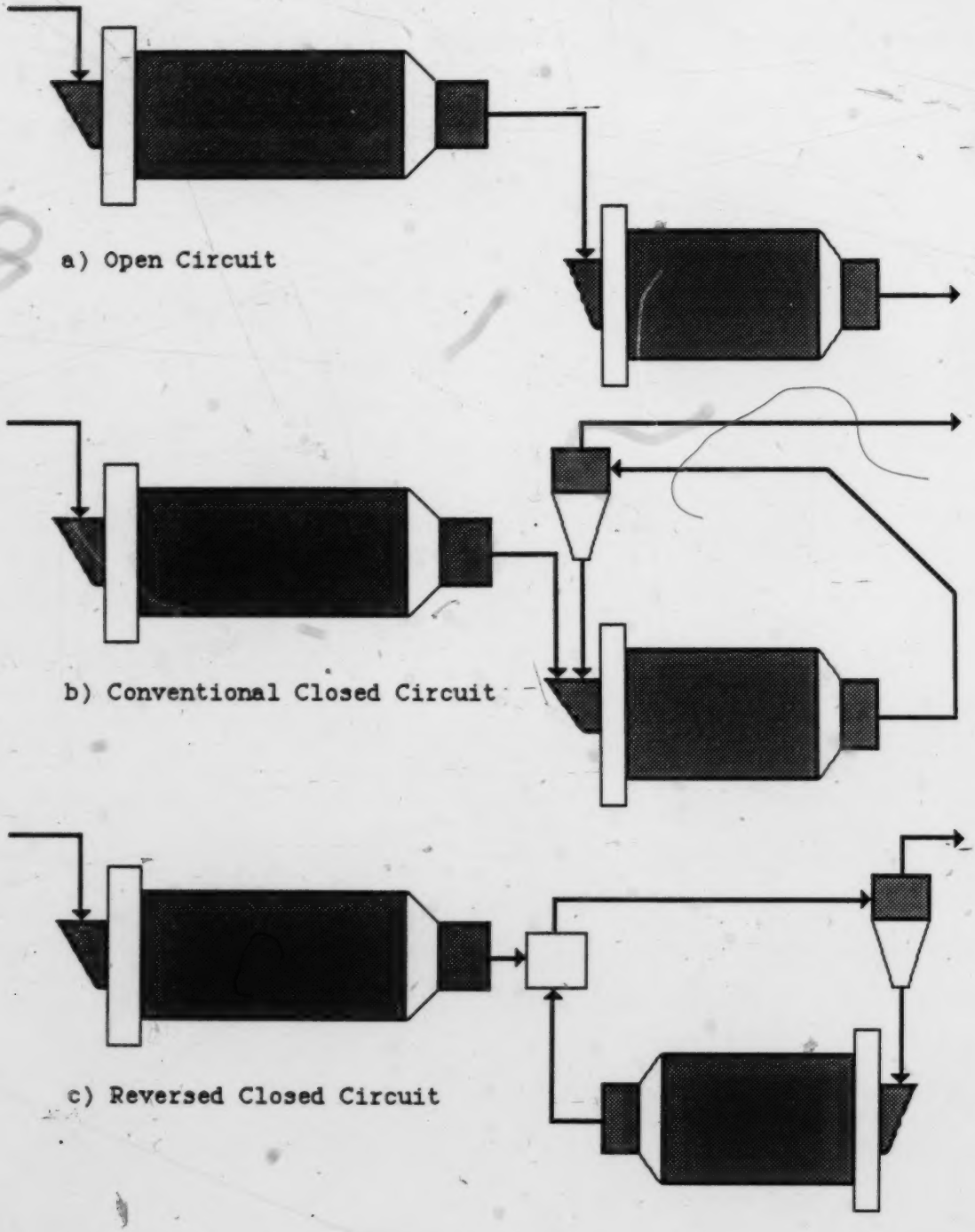


Figure 3.1 Grinding Circuit Configurations Currently Included in the Simulator

focus of all subsequent work involving both open and closed-loop simulations. The former two are included simply for demonstration and educational purposes to graphically indicate the limitations of these configurations. The mill models used in the simulator have been shown to be effective in modelling rod and ball mills only, a point that the potential user is asked to keep in mind. The form of the grinding mill models used in this work follow those outlined by Austin (1972), which are known as time continuous, discrete size interval models. This form of the grinding equation is the basic framework of the unit operations approach to modelling grinding mills, and has been found to be effective for a wide range of grinding mills.

By far the most popular classifying device in mineral processing is the hydrocyclone classifier due to its low capital and maintenance costs and its versatility. The most common application of the cylone is in providing some positive control of the product size distribution from a grinding circuit. From a simplistic point of view, the classifying action within the cyclone is due to a balance between a centrifugal force that moves larger or higher density particles to the underflow stream, and a drag force that transports the smaller or lower density particles to the overflow stream. However, a multitude of other factors come into play that preclude the development of good mechanistic or theoretical models to describe the separating mechanisms. As such, phenomenological modelling based on the

theories of fluid flow and boundary layer equations have, to date, met with little success. As a result, various workers, most notably Plitt (1976) and Lynch and Rao (1975) have utilized extensive databases for a wide range of cyclones to develop empirical correlations between the independent design and operating variables and the cyclone performance indices.

Smith and Guerin utilized the models developed by Lynch and Rao which were found to be applicable to their situation, as well as others (Draper et al, 1969). Since not enough information is available to use Plitt's models, it was decided to retain those outlined by Smith and Guerin. If the reader is interested in using the work of Plitt, those equations have been combined in a stand alone subroutine available with the SPOC software (Laguitton 1985). Substituting the so called HCONE subroutine for the CLASS subroutine used in this work would require some slight modifications, but is not considered to be that difficult, being mainly that of changing the subroutine block format and variable names in order to be compatible with the structure of the simulator.

The remaining units of any given grinding circuit can be loosely categorized as conveyors, pipes, sumps and pumps. Conveyors and pipes are an important consideration in simulation studies as they act as fixed or variable time delays in the circuit. The importance of including time delays in a simulator used for control studies cannot be over emphasized as they are the major contributors to stability problems and

invariably result in a damping of the control action. The sump is modelled as a single perfect mixer, with the appropriate calculations being made to monitor the hold up of individual size classes, solids and liquids within the sump. Various options are available to model the pump, ranging from the steady state model which simply sets the volumetric flow rate out equal to that flowing into the sump, to the fixed and variable speed models developed using manufacturer's head capacity curves.

Finally, in order to handle situations in which more than one stream flows into a unit operation, blending subroutines have been included to avoid repetitive coding at the front end of the unit models. This improves the modularity of the simulator, and allows each unit to be modelled as a single input system. There are no dynamics involved in such a device, and the algorithm simply calculates the algebraic sum of the components flowing into the blender.

3.2 Simulator Models

The last two decades have seen an incredible growth in the amount of work performed in the area of modelling unit operations in the processing industries. As indicated earlier, the mineral processing industry has traditionally lagged behind, due in the most part to the lack of understanding of the unit operations involved. This problem in turn is compounded by the complexities of the models that have evolved from the study of the unit operations in mineral processing.

Early work, at both the plant and research level, involved attempts to produce empirical models from operating data. Although effective for specific plants, given quite a narrow operating range, such models had severe limitations and never gained widespread acceptance. More recently though, phenomenological models, in the case of grinding mills, and empirical models of the hydrocyclone have been developed and refined such that they are valid over a wide range of operating conditions. Other models used in the simulator have been developed at the University of Alberta in recent years borrowing generously from the experience of individuals in the field of chemical engineering. As a result, the models used in this simulator are considered as the best that are currently available. Given the structure of the simulator, it is felt that unit modules can be easily inserted into the program by individual users. As such, the author is convinced that if the user is unsatisfied with any of the models currently in place, with very little time and effort, an alternative model can be substituted into the simulator.

3.2.1 Mill Model

The basic form of the mill model used, in matrix notation, is attributed to Kelsall et al. (1969) and has been found to be applicable to a wide range of industrial grinding mills.

$$\dot{\underline{p}} = [(B - I)S - \frac{1}{\tau}I]\,\underline{p} + \frac{1}{\tau}\,\underline{f} \qquad (3.1)$$

where:

B = sub-diagonal matrix of breakage elements
S = diagonal element of selection rate constants
I = identity matrix
τ = grinding mill (mixer) residence time
$\underline{p}$ = vector of size fractions in the mill (mixer) product stream
$\underline{f}$ = corresponding vector in the mill (mixer) feed stream

This model assumes perfect mixing, the breakage rate of ore particles is a function of particle size only, the ore consists of only one component and that the slurry volumetric holdup is constant.

This model can be improved by considering that the ore is composed of more that one component, e.g. a hard and a soft component, with each component having unique breakage and selection functions, density and size distributions. Small industrial or pilot plant scale grinding mills can be adequately modelled as single perfect mixers, whereas Stewart and Weller (1977) have indicated that it is more appropriate to model larger mills as up to three perfect mixers in series. As such the grinding mill model is developed so as to account for the multiple mixers in series. To facilitate the understanding of this model it is appropriate to expand equation 3.1 and only consider the i^{th} size fraction of the k^{th} component in the m^{th} mixer, such that:

$$\frac{d(V_k^m \rho_k p_{ik}^m)}{dt} = qF_k^m \rho_k f_{ik}^m - qP_k^m \rho_k p_{ik}^m - s_{ik}^m V_k^m \rho_k p_{ik}^m + \sum_{j=1}^{i-1} b_{ijk}^m s_{jk}^m V_k^m \rho_k p_{jk}^m \qquad (3.2)$$

where:

V_k^m = volumetric holdup of component k in mixer m
q = volumetric flow rate into and out of mixer m
p_{ik}^m = mass fraction of size class i, component k, in mixer m product
f_{ik}^m = mass fraction of size class i, component k, in mixer m feed
F_k^m = volume fraction of component k flowing into mixer m
P_k^m = volume fraction of component k flowing out of mixer m
s_{ik}^m = selection element for size class i, component k, mixer m
b_{ijk}^m = breakage element, fraction of size class j, of component k, that reports to size class i after breakage in mixer m
ρ_k = density of component k

Given that

$$V_k^m = P_k^m \; V^m$$

where, V^m is the volume of mixer m, and is assumed to be constant, that for any given component, ρ_k is constant, and

at any instant in time the volumetric flow rate is also constant, then equation 3.2 can be rewritten as:

$$\frac{d(P_k^m p_{ik}^m)}{dt} = \frac{q}{V^m}(F_k^m f_{ik}^m - P_k^m p_{ik}^m) - s_{ik}^m P_k^m p_{ik}^m + \sum_{j=1}^{i-1} b_{ijk}^m s_{jk}^m P_k^m p_{jk}^m \qquad (3.3)$$

The differential on the left hand side of equation 3.3 can be broken down to its constituent parts,

$$\frac{d(P_k^m p_{ik}^m)}{dt} = P_k^m \frac{d(p_{ik}^m)}{dt} + p_{ik}^m \frac{d(P_k^m)}{dt} \qquad (3.4)$$

The second differential on the right hand side of equation 3.4 is related to the macro solids balance around mixer m, once again assuming perfect mixing, i.e.

$$\frac{d(P_k^m)}{dt} = \frac{q}{V^m}(F_k^m - P_k^m) \qquad (3.5)$$

Substituting equation 3.5 into 3.4, then 3.4 into 3.3, moving the second term of the left hand side of the revamped equation 3.3 to the right hand side, dividing through by P_k^m, and realizing that

$$\frac{V^m}{q} = \tau^m$$

yields

$$\frac{d(p_{ik}^m)}{dt} = (f_{ik}^m - p_{ik}^m) \frac{F_k^m}{P_k^m \tau^m} - s_k^m p_{ik}^m + \sum_{j=1}^{i-1} b_{ijk}^m s_{jk}^m p_{jk}^m \quad (3.6)$$

It is equation 3.6 which forms the basis of the grinding mill subroutine used in the simulator. This algorithm is included in the subroutine MILL which requires as arguments simply the labels of the input and output stream and the mill identification number used to differentiate between the rod mill and the ball mill.

For each integration step through the simulator there are max(m·k·i) differential equations to be evaluated for each mill for the comminution process alone. In addition to this, there are the two component macro balances (as per equation 3.5) to be solved. There is also the water balance equation, but given a constant mixer volume, this equation can be solved by difference. As a result of this computational load it is readily apparent that the simulator will run fastest for a single component feed using single mixer mill models. Once the user indicates a two component feed and a three mixers in series mill model for a feed with sixteen size classes there are almost one hundred differential equations to be solved for each mill model at each iteration. This increases the computational costs proportionately and may be a factor to consider for users involved in time sharing or multi-user systems. The model indicated in equation 3.6 has been tested extensively, and the MILL

subroutine in which it has been used has been shown to be valid for mill models ranging from one to three perfect mixers.

The aim of including a multi-component stream structure is to facilitate modelling hardness changes while the simulator is running. The previous work by McDougall achieved this by simply multiplying the selection elements in the rod and ball mill by a constant. As such, the selection elements in the rod mill were altered, then, after a suitable time delay, the ball mill selection elements were changed accordingly. This method has obvious limitations, i.e. the time delay has to be chosen *a priori*, and cannot be adjusted once the simulator is started to account for changes in flow rates. By modelling and accounting for each component separately the user doesn't have to worry about time delays between mills as that will be taken care of simply as each component flows through the circuit. The user can specify both the time and rate at which the component ratios in the feed material change.

A step change from 100% component 1 to 100% component 2, although probably not that realistic, but useful for open loop studies, reveals a numerical problem in the algorithm that the user should be made aware of. Examination of equation 3.6 reveals that the first term on the right hand side is multiplied by the ratio of the volume fraction of component k flowing into the mixer at any time t, and the volume fraction of component k already in the mixer at that

time. The algorithm is forced to avoid making a division by zero in the case where there is no component k in the mixer. The problem is encountered, however, when there is no component k in the mixer, yet the volume fraction of component k entering the mixer is unity, i.e. immediately after a step change in the component ratios. The volume fraction of component k is calculated by integrating equation 3.5, and as such equations 3.5 and 3.6 are said to be coupled. Problems are encountered when the time constants of coupled differential equations are widely different, such as is the case with this algorithm. The time constant of equation 3.5 is simply q/V^m, and is relatively constant. On the other hand the coefficient of the first term on the right hand side of equation 3.6, which in itself is a component of the time constant, is the ratio of component k in the feed to component k in the mixer, i.e. $F_k^m/P_k^m\tau^m$. Given that τ^m is virtually constant, it can be seen that this ratio can vary from zero to infinity. The ratio of the magnitude of time constants of coupled differential equations is known as the stiffness ratio, and the value of this ratio has been shown to have a marked effect on numerical stability of large systems of nonlinear equations. Although the DYFLO integration subroutines used in the simulator offer the option of using first, second and fourth order methods of solving differential equations, it can be shown that acceptable accuracies for stiff systems can be achieved with the first order method (Franks, 1972). In fact, the first order method is more

efficient than the second or fourth order methods by a ratio of 2 and 4 respectively. As a result of these observations, the first order, or simple Euler, method of solving differential equations has been used in the simulator. Previous work by McDougall has indicated that algorithms developed by Franks specifically to handle stiff systems (Franks, 1982), do not, in this particular application, allow one to increase the integration step size sufficiently to offset the increased computational costs at each incremental step. If the user is not satisfied with this arrangement, it is possible to alter the source code of one of the initialization subroutines, and by changing the value of one of the DYFLO counters, implement a second or fourth order integration algorithm.

Even with this precaution in effect, it is possible for the algorithm to calculate a negative value in the coarsest size fraction immediately after a step change in the component ratios, obviously an unreasonable situation. The errors are manifested in the pan fraction since it is calculated by difference. If one inspects the mill discharge at each time increment, the negative fractions are apparent for one or two iterations after the disturbance occurs. However, if one sums the four or five finest size classes, e.g. to monitor the -150 micron material, the magnitude of the erroneous pan fraction is minimal. As a result, it is felt that although the algorithm is not perfect in the theoretical sense, it is totally suitable for its current

application.

It is appropriate at this time to briefly discuss the form of the breakage and selection functions used by the simulator. Further information on developing grinding mill models can be found in Lynch (1977) or Laguitton et al (1984). Smith and Guerin found that a simple two parameter, normalized Gaudin-Schuhmann cumulative distribution for the breakage products,

$$B(x,x_0) = k\ (x/x_0)^{\alpha}, \qquad \alpha>0 \tag{3.7}$$

was adequate for describing the breakage of particles from initial size x_0 to a range x. Given that the size classes used form a geometric series with a size class ratio of $1/\sqrt{2}$ then the discretized breakage function is given by

$$b_{ij} = B_i - B_{i+1} \tag{3.8}$$

where

$$B_i = \left[\frac{1}{\sqrt{2}}\right]^{((i-j-1)\alpha)}$$

Noting that for $i \leq j$, $b_{ij}=0$, since it is assumed that there is no particle agglomeration and no daughter particles remain in the original size class, then the final form of the discretized breakage function is

$$b_{ij} = [1/\sqrt{2}]^{(i-j-1)\alpha} - [1/\sqrt{2}]^{(i-j)\alpha}, \qquad i>j \tag{3.9}$$

Experimental work by Smith and Guerin indicated that for the rod mill, α=0.676, and for the ball mill, α=0.652.

Similarly, for the selection rate constants, Smith and Guerin found the following functional forms to be quite satisfactory, with x_i being the characteristic size expressed in microns. For the rod mill:

$$s_i = 0.0212 \qquad x_i \leq 1000\mu$$
$$s_i = 0.0212(x_i/1000)^{4.48} \qquad x_i > 1000\mu \tag{3.10}$$

And for the ball mill:

$$s_i = 1.045(x_i/1000)^{1.818} \tag{3.11}$$

To simulate a harder ore these selection elements are simply multiplied by 0.5 with the breakage elements remaining unaltered.

Various other functional forms have been discussed in the literature, eg. Rajamani (1979) uses a selection function that is dependent on mill power, mill solids holdup and percent solids, of the form

$$s_i = \frac{s_i^E P}{F H} \tag{3.12}$$

where:

s_i^E = a constant characteristic of the material
P = the mill power
F = the mass fraction of solids in the mill
H = the total mass holdup in the mill

It is beyond the scope of this work to determine the strength and weaknesses of various functional forms of the selection and breakage functions. Suffice to say, Smith and Guerin found these models to be suitable for their purposes, and as such, they are suitable for the purposes of this author.

3.2.2 Hydrocyclone Model

As indicated earlier, the empirical equations of Lynch and Rao were found to work in the application outlined by Smith and Guerin. Since these models have been taken from the work of Smith and Guerin, they are simply listed here as presented in that work.

$$Q = 32.42 * P^{1/2} * F_{wf}^{1/8} \tag{3.13}$$

$$W_o = -9.13 + 0.752 * W_f \tag{3.14}$$

$$\log_{10}(d_{50c}) = 2.87 + 0.0145 * P - 0.0202 * W_o \tag{3.15}$$

$$Y_c(d) = \frac{e^{\alpha x} - 1}{e^{\alpha x} + e^{\alpha} - 2}, \quad \alpha = 0.89, \quad x = d/d_{50c} \tag{3.16}$$

$$R_f = 1 - \frac{W_o}{W_f} \qquad (3.17)$$

$$Y_a(d) = R_f + (1 - R_f) * Y_c(d) \qquad (3.18)$$

where:

Q = the volumetric flow rate of slurry to the cyclone (m^3/h)

P = the cyclone inlet pressure (kPa)

F_{wf} = mass fraction of water in the cyclone feed slurry

W_o = the mass flow rate of water in the overflow product (tph)

W_f = the mass flow rate of water in the cyclone feed slurry (tph)

d_{50c} = the corrected equiprobable partition size based on classification alone

$Y_c(d)$ = the mass fraction of classifier feed solid material in size class (d) that reports to the underflow product by classification alone, i.e., excludes short circuiting

R_f = the mass fraction of water in the cyclone feed recovered in the cyclone underflow product

$Y_a(d)$ = the mass fraction of classifier feed solid material in size class (d) that actually reports to the underflow product

The dynamics of the hydrocyclone can be ignored since the residence time within the cyclone is small compared to the other process units. Subroutine CLASS models the cyclone and requires only the labels of the feed, underflow and overflow streams in the argument string. If other classification

devices are to be modelled in the circuit it is a simple matter to replace CLASS with the appropriate algorithm.

Since the simulator has been expanded to handle two solids components, consideration has to be given to the situation in which the two components have widely different densities. This is a particular problem in processing any ore where there are two, fairly well liberated, components of different density, such that the high density material preferentially reports to the underflow stream. This results in overgrinding of the high density component and potential reduced recovery in the flotation circuit, (e.g. Ramirez-Castro and Finch, 1980) This problem has been investigated by various workers, (Finch & Matwijenko, 1977, and Lynch, 1977), and has resulted in a general model that relates d_{50c} to the difference in densities between the solid fraction, ρ_s and the supporting fluid, ρ_l.

$$d_{50c} \propto (\rho_s - \rho_l)^{-K} \qquad (3.19)$$

Lynch (1977) indicates that for similar minerals and cyclones K = 1.0, whereas Finch and Matwijenko found K to vary from 0.75 to 1.5 depending on the ore in question. Since this sort of relationship would appear to be site and ore specific, the author leaves it up to the individual to determine the exact relationship for his/her particular application. For the sake of simplicity, the relationship currently used in the simulator is

$$C.F._{,} = \frac{(\rho_{c1} - \rho_l)}{(\rho_{c2} - \rho_l)} \qquad (3.20)$$

Where C.F. simply indicates that this is a correction factor to adjust the d_{50c} calculated using equations 3.13 to 3.15, with the subscripts c1 and c2 denoting component one and component two respectively in the two component solids fraction. Comparing equation 3.19 and 3.20, it is apparent that the numerator of 3.20 is simply a constant of proportionality such that for $\rho_{c1}=\rho_{c2}$ the correction factor equals unity. The value of the exponent K in equation 3.19 is set to unity in equation 3.20 since this is the power as predicted by Newton's law of settling and it also appears to be a good average figure based on published information. Using a correction factor of the form of 3.20 is reflected in a shift of the partition curve of the heavier component to the left as seen in Figure 3.2. Although this representation may be slightly idealized, and not absolutely correct, its inclusion is important. By incorporating the differences in hardness of the two components, as well as the differences in densities, the simulator will demonstrate that the size distribution of the softer, denser mineral in the product stream will be much finer than that of the lighter, harder mineral. This can best be demonstrated by presenting the results of a particular simulator run, Figure 3.3, in which the simulator processed a feed stream of 50% soft dense material and 50% hard light material until an approximate steady state was achieved. The important note to be made

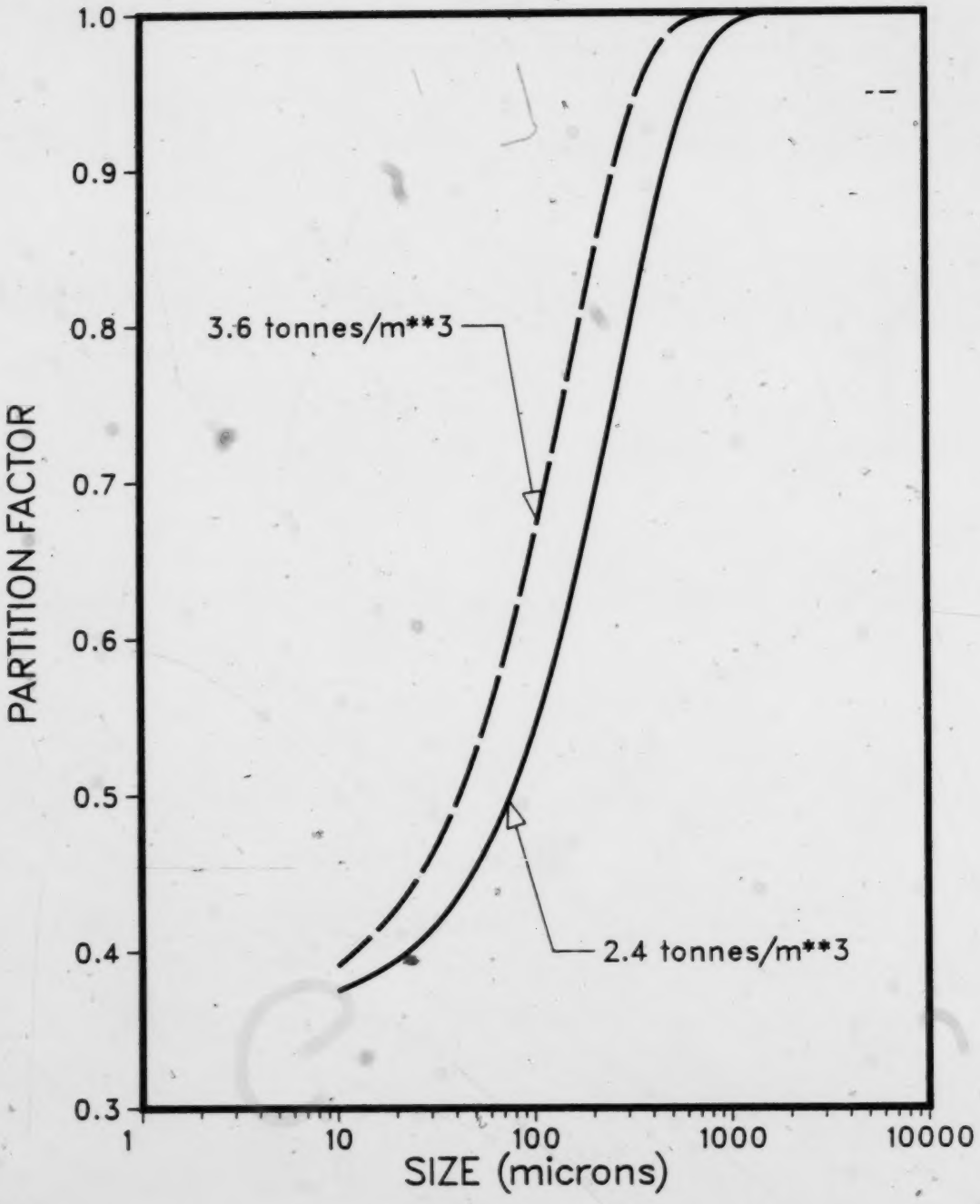


Figure 3.2 Partition Curve as Predicted by the Simulator Hydrocyclone Model for Components of Differing Densities

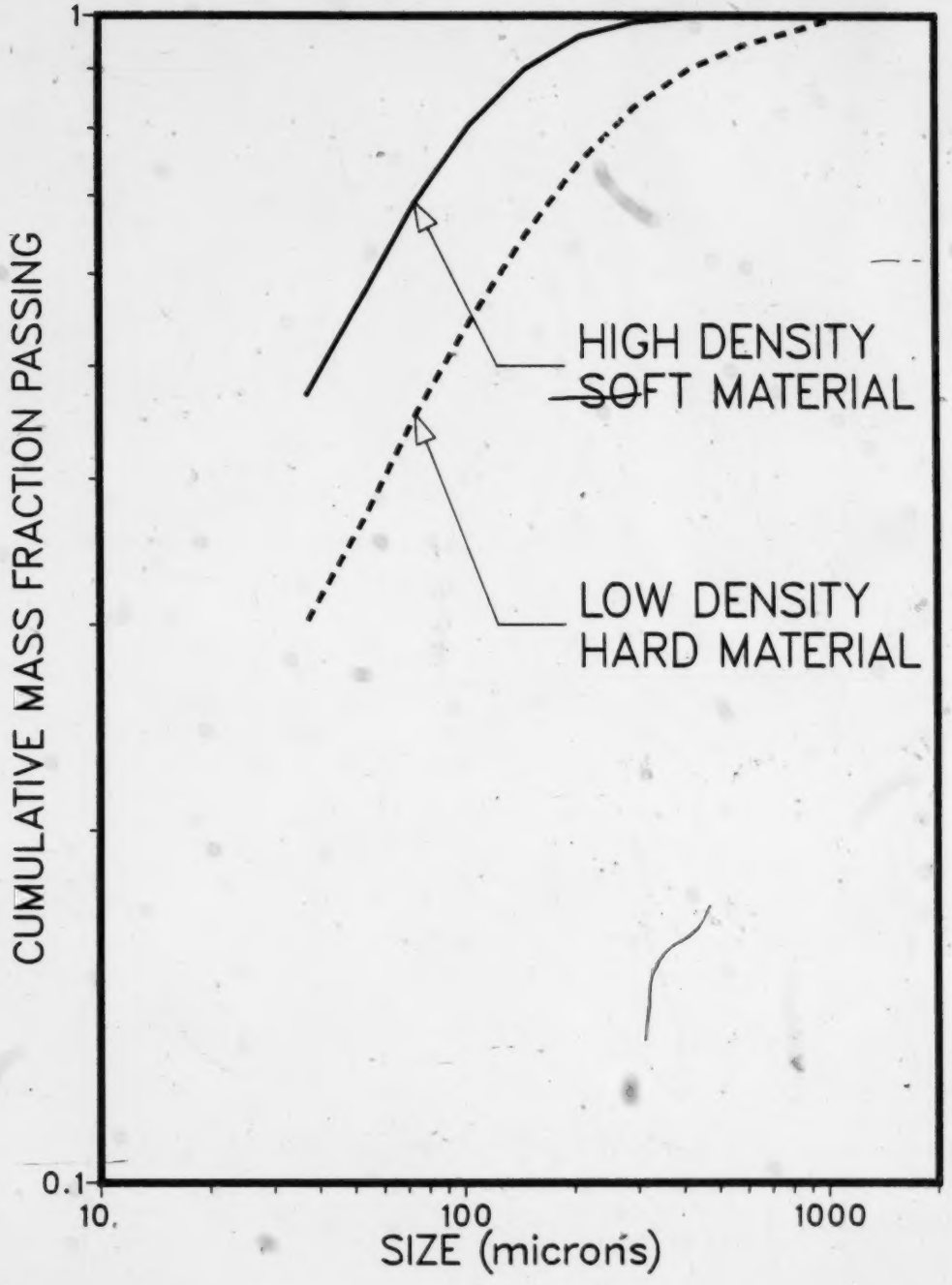


Figure 3.3 Simulator Generated Output Comparing the Product Stream Size Distribution of a Soft Heavy Component to that of a Hard Light Component

here is that such results, although indicative of the real situation, can only be analyzed on a qualitative basis since the simulator has not been verified with any industrial data apart from those limited results available through the work of Smith and Guerin.

3.2.3 Fixed Time Delay

The rod mill feed conveyor can be accurately modelled as a fixed time delay. The algorithm used in the simulator is based on one suggested by Franks (1972) with suitable modifications made to handle the multi-component stream attribute structure, e.g. the size distributions of each component and the mass and volumetric flow rates. Since there is little or no mixing along the length of the conveyor, the feed material added to the belt at each time interval can be modelled as a pseudo discrete unit. This sort of system can be handled quite efficiently on a digital computer by using a circular buffer as shown in Figure 3.4. Conceptually, the subroutine allocates N spaces to the conveyor, where N is the conveyor time delay divided by the integration interval, rounded off to the nearest integer. Instead of moving the stream attributes down through the registers at each time interval, it is more efficient to simply increment a single counter, which in effect rotates the buffer, and withdraw the appropriate cell according to the value of the counter at that particular time interval. To facilitate bringing the simulator to steady state when

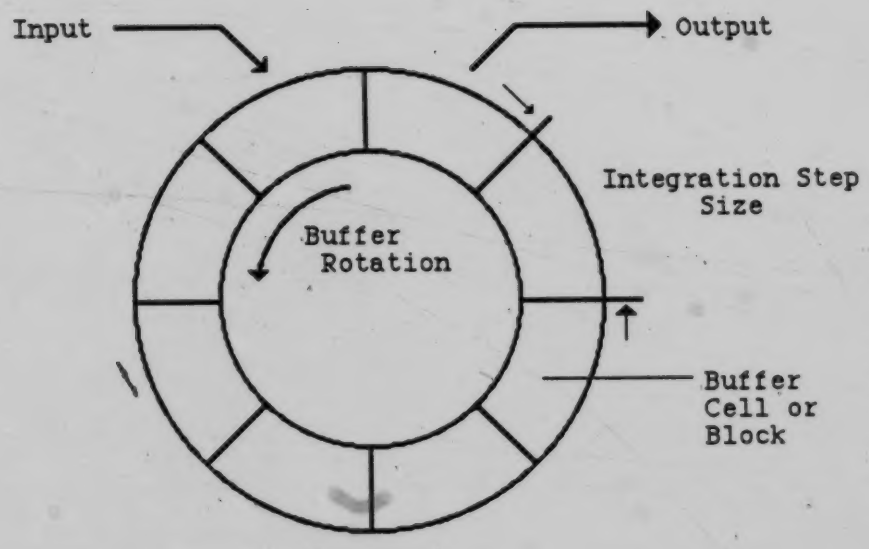


Figure 3.4 Circular Buffer: with no movement across cell boundaries

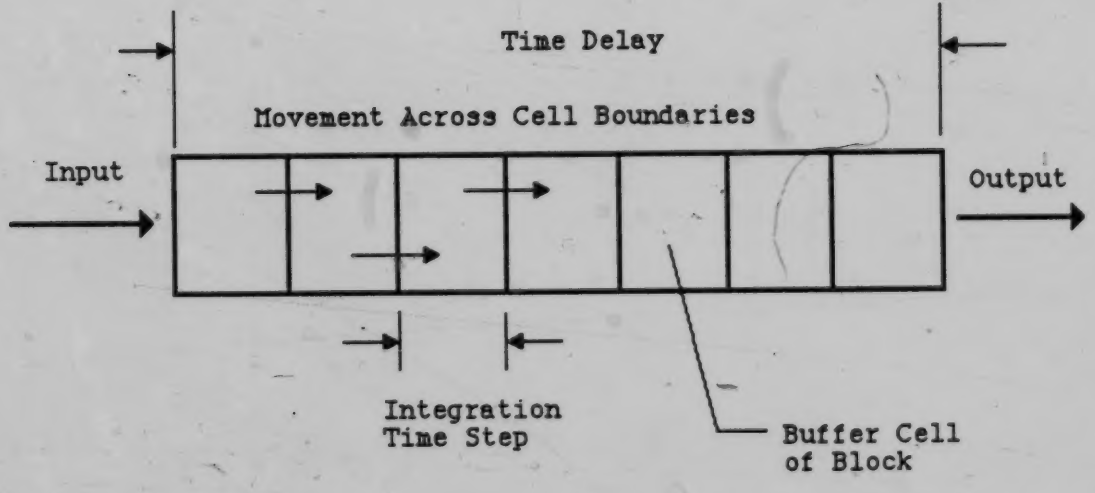


Figure 3.5 Straight Line Buffer: with movement across cell boundaries

started from zero initial conditions, and to avoid having to store the conveyor cell data from steady state simulations at the end of each run, all the cells of the conveyor are filled with the feedstream material during the first call to the subroutine. This is analogous to considering the conveyor to be loaded with material at the beginning of each run, which given the nature of mineral processing operations, appears to be quite realistic.

3.2.4 Variable Time Delay

It is well recognized that time delays in a circuit can lead to control problems, but if compensated for, can be dealt with by slowing down the control action. This statement is based on the premise that the time delay can be measured and is fairly constant during the control cycle. Problems can develop when the time delay varies according to the volumetric flow rate through a given unit, making it imperative that some form of variable time delay is included in a dynamic simulator. Flintoff et al (1985) recognized this requirement and developed an algorithm based on some documentation relating to ACSL (Advanced Continuous Simulation Language), a simulation package that has found extensive use in the Department of Chemical Engineering at the University of Alberta, (ACSL). The algorithm was documented by Flintoff et al (1985) and McDougall (1987), but for reasons of completeness it is necessary to describe it again here, although little of the algorithm has been changed. The model

can be conceptualized as a straight line buffer as shown in Figure 3.5, with movement of material across the boundaries of the cells. The first cell is filled from the input stream, once it is full, the attributes of each cell are shuffled down the buffer. Any residual material from a given time step is then placed in the now empty first cell. Thus for N cells in the buffer, the shuffle temporarily creates an $N^{th}+1$ cell, which, along with the N^{th} cell, holds the material that is introduced to the pipe outlet stream. To simulate the continuous nature of a real pipe, the attributes of the outlet stream are determined from a linear combination of the last two cells in the buffer. The relative weighting of the two blocks is related to the residual amount of material in the first cell. This algorithm has been found to be effective in modelling a variable time delay, and its use is quite straightforward. The user only has to specify the input and output stream labels, the total volume of the VTD and the number of blocks or cells included in the buffer. The rule used to calculate the number of blocks, as suggested by Flintoff et al. is as follows,

$$2 \leq N \leq \frac{V}{3Q_m \Delta t} \qquad (3.21)$$

where:

V = Variable time delay holdup, m^3

N = Number of blocks or cells in the buffer

Q_m = the maximum volumetric flow rate expected in the variable time delay feed stream, m^3/hour

Δt = the integration step size

with N, in general, being chosen as large as possible. As is always the case with discretization, there is some loss of accuracy, but this algorithm provides a good compromise between computational efficiency and model accuracy.

For increased computational efficiency the variable time delay is also used to model the plug flow element in the grinding mills. Although it is important to consider this component of the mill model as a variable time delay, from a physical point of view it would seem to be more appropriate to think of it as an open launder rather than a pipe. The difference here is important: for a pipe, given that it is completely full, the flow rate out has to equal the flow rate in due to the incompressibility of the pulp. On the other hand, with an open launder, given a step change in the volumetric flow rate into one end, there is a definite time delay until that change in flow rate is realized at the discharge end, a situation that can best be visualized as a wave moving down the launder. This results in a slight modification to the VTD algorithm. A flag called IPIPE is included in the argument list, if the VTD follows a mill model, i.e. it is modelling the plug flow, then IPIPE is set to unity, otherwise the VTD is assumed to model a pipe and IPIPE is set to zero. Thus, in the algorithm, if IPIPE=0 the volumetric flow rate out of the VTD is set equal

to the volumetric flow rate into the pipe. If IPIPE=1 the volumetric flow rate out of the VTD is calculated as a linear combination of the flow rate information carried in the last two cells of the straight line buffer. It appears important to make this differentiation because otherwise changes in volumetric flow rates appear to be transmitted through the circuit too quickly. For example, given that the mill models assume an overflow discharge, then any change in the volumetric flow rate into the mill is immediately reflected in the mill disharge. This discharge, with the altered volumetric flow rate, is then fed into the VTD model, which if modelled as a pipe, immediately produces a change in the volumetric flow rate out of the VTD. In such a case, a change in the flow rate into the mill causes an immediate change in the flow rate out of the mill, a situation that wouldn't appear to reflect what occurs in a real situation.

3.2.5 Sump and Pump Models

The sump model used in the simulator is basically unchanged from the one used previously and is found in the SMPMOD subroutine. It consists of a series of differential equations to calculate the solids mass holdup of individual size fractions and the overall solids and water mass holdup within the sump. Since the slurry density has an effect on the pump head, and since the simulator can now handle solids components of differing densities, it is necessary to

calculate the average slurry density in the sump. Once all the sump calculations have been made, a user specified pump algorithm is called in order to calculate the flow rate out of the sump. The sump inventory is then recalculated by subtracting the volume of material removed by the specified flow rate in a single time interval. Various flags have been included to indicate sump overflow, low sump level and empty sump conditions to the user during the simulation. The fact that the overall module operates by first adding the input stream, then calculating the pump action, and finally subtracting the output stream reveals a weakness in the subroutine, a weakness that is apparent whenever a continuous process is simulated on a digital computer. At any point in time, there is a continuous flow into the sump, and a continuous flow out of the sump. By discretizing the system, some accuracy is lost. However, any improvements to the current subroutine would certainly impart some increased computational costs. The pump calculations are an iterative numerical process, and in part rely on the static head in the sump. If one were to consider the inlet and outlet flow rate simultaneously while calculating the pump equations, then the iterative process would be further expanded in order to calculate the pumping rate at which flow in, flow out, sump level and pumping rate were all compatible. Although some investigation into this problem is warranted, the author feels at this time that the accuracy of the model is adequate for the simulator requirements.

The simulator provides four options to calculate the flow rate out of the sump, the pump algorithms are found in the PUMP subroutine which is called directly from the SMPMOD subroutine. The first option, and the most primitive, simply sets the volumetric flow rate out of the sump equal to that flowing in. This simulates perfect sump level control, but can only be implemented once the circuit is at steady state. The second option is only slightly more sophisticated, calculating the outlet flow rate as a function of the pump level in the sump. This is based on the simple first order differential equation that describes the pulp level as a function of the flow in and the flow out

$$\frac{Adh(t)}{dt} = F_{in} - F_{out} \qquad (3.22)$$

Given that the flow rate out can usually be described as a function of the square root of the height then Eq. 3.22 can be rewritten as

$$\frac{dh(t)}{dt} = \frac{1}{A} \left[F_{in} - K\sqrt{h} \right] \qquad (3.23)$$

where K is known as the discharge coefficient. At steady state,

$$\frac{dh(t)}{dt} = 0$$

which implies that

$$F_{in} - K\sqrt{h} = 0$$

or conversely

$$K = \frac{F_{in}}{\sqrt{h}} \qquad (3.24)$$

As a result, it is apparent from Eq. 3.24 that given the steady state sump level and steady state flow rate into the sump the valve constant can be calculated. Thus the user only has to specify these two parameters, plus the appropriate flag to call the algorithm, in order to run the simulator. Although this type of calculation is not recommended for any in-depth circuit analysis, it may be of some use in those cases in which the pump characteristics are unknown or not considered to be of importance to the simulation.

A third pump option, and the one used by Flintoff et al, is a fixed speed pump. Using the manufacturers pump head-capacity curves, a functional form for a 10 in. x 8 in. S.R.L. pump operating at 700 rpm was developed to express total dynamic head (TDH (m)) as a function of volumetric flow rate (Q (m^3/h)).

$$TDH = 2.193 - 3.414*10^{-3}\ Q - 1.454*10^{-4}\ Q^2 \qquad (3.25)$$

The limitation of using a single speed pump can be easily demonstrated as in Figure 3.6. A 20% increase in the mass flow rate of water to the sump, in an open loop simulation,

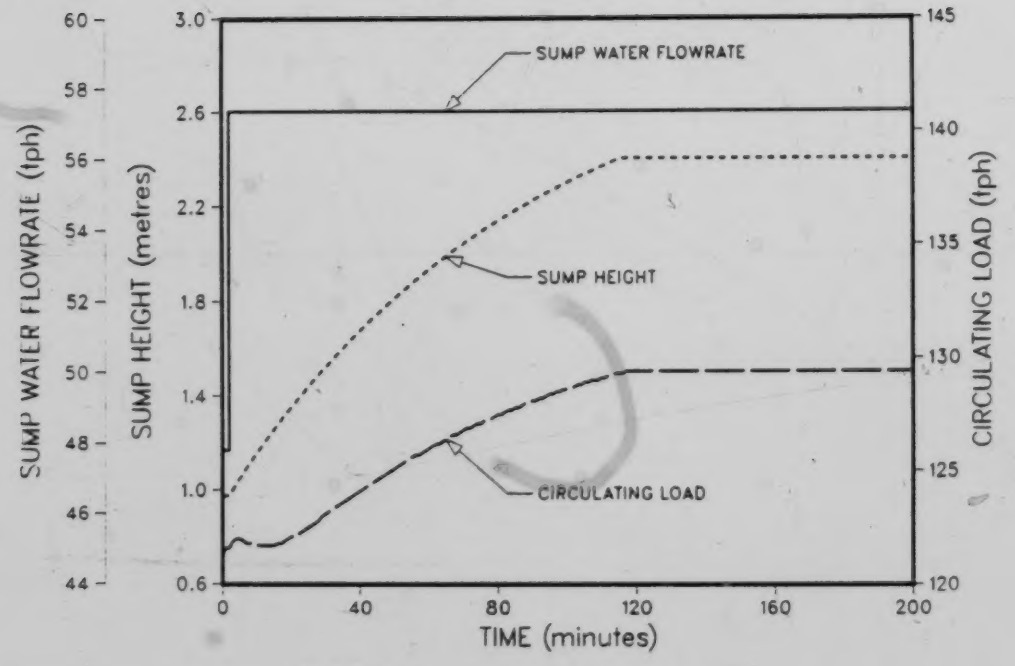


Figure 3.6 Open Loop Simulation of a 20% Increase in Sump Water Flow Rate at t=2 Minutes using a Single Speed Pump

indicates that after almost 120 minutes the sump overflows which in turn affects the circulating load. Although this may be a problem unique to this simulator alone, it does highlight a potential problem for future control studies.

Recognizing that a single speed pump has some inherent limitations, a variable speed pump model is included as the last option. The advantage here is that a simple feedback controller can be implemented to manipulate the pump speed to control the pulp level. The inclusion of a variable speed pump is also important in that various authorities in grinding circuit control, e.g. Gault et al. (1979), Lynch (1977), state that effective control of short term fluctuations in grinding circuits is not possible with a fixed speed pump. The inclusion of the variable speed pump model offers the oppurtunity to study and evaluate such configurations. Once again the manufacturers head-capacity curves for a 10 in. x 8 in. SRL variable speed pump were used to develop the model. By digitizing the curves for different speeds and volumetric flow rates, and using a stepwise linear regression package available at the University of Alberta, (BMDP, 1985), a fairly accurate functional form has been developed relating TDH to pump speed and volumetric flow rate.

$$TDH = -2.6954 - 18.4936 \left[\frac{Q}{1000}\right]^2 + 0.5109 \left[\frac{RPM}{100}\right]^2 \quad (3.26)$$

Figure 3.7 shows the good correlation between the digitized points used for the regression and the model prediction as

given by Equation 3.26. A discussion of the sump level controller is included in the subsequent chapter.

For both of the pump models the volumetric flow rate must satisfy the mechanical energy balance over the sump-pump-cyclone system, i.e. the total dynamic head (TDH) must equal the resistance dynamic head (RDH). The resistance dynamic head model, as described by Flintoff et al. (1985), is composed of the net static head, pressure head and friction head contributions, such that

$$RDH = (12.2 - L) + \left[\frac{0.102}{\rho}\right]\left[\frac{Q}{32.42\ F_w^{0.125}}\right]^2 + 3.44\times10^{-4}\rho\ Q^{1.85} \qquad (3.27)$$

where:

12.2 = lift from pump discharge to cyclone inlet, m
L = sump level measured against pump discharge, m
ρ = cyclone feed pulp density, t/m^3
Q = volumetric flow rate, m^3/hr
F_w = mass fraction of water in cyclone feed slurry

Equating equations 3.27 and 3.26 or 3.25, given all parameters except Q, the pump algorithm incorporates the Newton-Raphson method to solve for Q at each iteration.

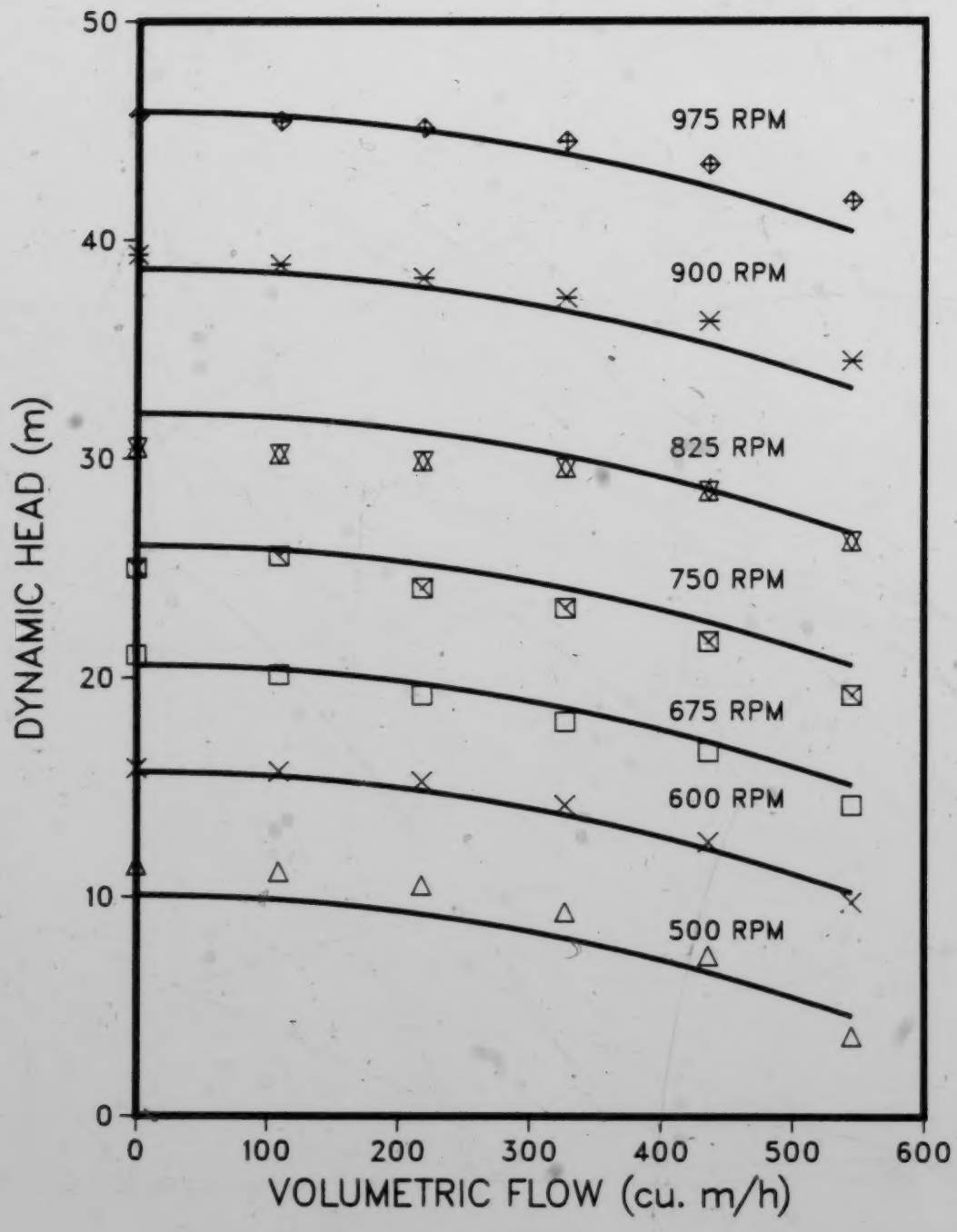


Figure 3.7 Correlation Between Pump Curves and Model Predictions for a 10" x 8" SRL Pump

3.3 Simulator Structure

It is appropriate at this time to present a fairly detailed discussion of the simulator itself, how it is put together and how the overall structure has been conceived in order to facilitate expansion by individual users. Probably the major emphasis in developing this simulator, apart from producing one that works, has been to develop a package that is not only easy to use, but is easy to alter and expand. As a result of this design criteria, the simulator is extremely modular, with the driver consisting of nothing more than a series of calls to various subroutines. Extensive use is made of COMMON BLOCKS in order to keep the argument strings of individual subroutines as short as possible.

A valid argument can be made for avoiding the use of common blocks since there is the ever present danger of inadvertently altering the wrong variable within a given common block. If this sort of error occurs, it can be an extremely time consuming procedure to track it down and make the correction. However, the only other option is to use extensive subroutine argument strings, and it has been this author's experience that it is just as easy to make errors in the ordering of the subroutine arguments as it is to inadvertently alter the wrong variable in the common block. As will be shown in greater detail below, the major simulator parameters are carried in six major arrays within a single common block called DATA1 with each array having a clearly identifiable name to avoid any confusion. A second common

block, called CINT, carries all the relevant information required by the DYFLO subroutines, while a third common block, CHAR1, is used to carry character strings since the FORTRAN compiler cannot handle character strings and numerical variables in the same common block. Finally, a fourth common block, CNTRL, is used to carry the variables used in the closed loop simulations. The driver also utilizes a BLOCK DATA subprogram in order to initialize the major arrays to zero prior to each execution of the simulator. This is done in order to avoid problems of picking up stray values in the arrays during multiple runs, and also of providing a common starting point for each run to facilitate error checking during the developmental work.

3.3.1 The Executive

The simulator executive, or driver, is a short piece of code consisting of a series of calls to various subroutines. Figure 3.8 is included to provide a schematic illustration of how the open-loop executive functions, with each sub-program represented by a named block and the double headed arrows representing the exchange of information between the different components of the simulator.

The source code of the driver can be easily modified to output particular information for a given run that would not normally be included in the simulator output. Keeping the majority of the subroutines fairly small means that the source code can be individually altered and re-compiled

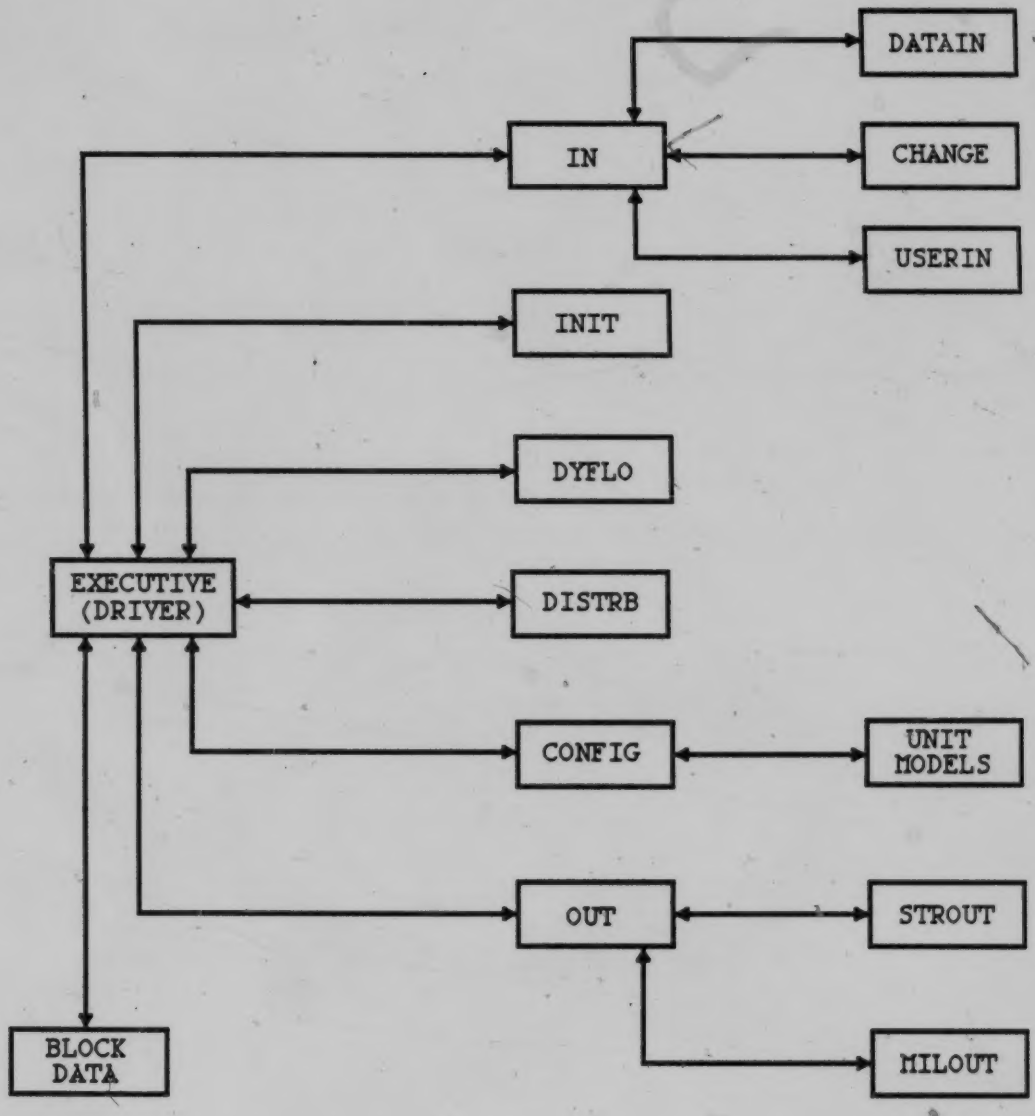


Figure 3.8 Schematic Diagram of the Open-Loop Simulator Structure

quickly and at very little cost. A second advantage in the modular structure is that it aids in the software debugging process since it is very easy to monitor variables before and after a given subroutine call in order to determine where a particular problem is occurring.

The first executable statement in the driver is a call to the subroutine IN which itself is another short piece of code. It is an interactive subroutine which allows the user various options for loading information into the simulator, either directly from a permanent data file, or interactively via the computer terminal. There are two types of data files available to load information to the simulator. The data file set up interactively by the user contains only the bare minimum information needed to start the simulator, this includes the individual model parameters and the feed stream array. All other stream arrays are set to zero by the block data subprogram and are not altered until the simulaton starts running. The second data file is the one that contains all the information generated by the simulator. It includes the complete data for all streams involved in the simulation so that once the information is loaded into the simulator the system is seen to be at either some steady state, or at least at a state equivalent to that in the simulator at the end of the run that created the data file. Subroutine IN offers three options for inputting information to the simulator:

1. use of an established data file that requires no changes to the information retrieved,
2. use of an established data file, but allowing the user the option of making a limited number of changes to the information retrieved, or
3. interactively set up a new data file.

If the first option is chosen, subroutine DATAIN is called which further prompts the user to specify whether the data file from which the simulator parameters are to be read is a complete file that has been previously generated by the simulator, or whether it contains only the information generated from the interactive program. Depending on the response a different set of READ statements are used. Subroutine DATAIN then returns control to the driver such that all pertinent information is now loaded into the appropriate arrays.

Subroutine CHANGE is called if the second option is indicated, with its first executable statement being a call to DATAIN in order to load the appropriate information. CHANGE then provides the user with a menu of possible options that can be changed at the beginning of the run. The most commonly used option is that of changing the simulator title so that the output data can be easily identified. Other options include changes to the equipment model parameters and the macro and micro properties of the feed stream, see Appendix A for a listing of the menus available in a typical

interactive session. This can be seen as an iterative process since the user can loop back to the menu as many times as necessary before returning control to the driver. This subroutine offers complete echoing of the input data to allow for error checking. In essence, this subroutine is very similar in nature to the interactive subroutine discussed below (option 3), and due to its interactive nature, and the extensive error checking facilities, is quite large. However, with the menu format, it is seen to be open to expansion, i.e. if more change options are required for a given application, it is only necessary to append the options to the menu and make the appropriate allowances with the statement numbers that control the program flow.

The third option results in a call to the subroutine USERIN, and as its name suggests, this option allows the user to interactively input the necessary information required to get the simulator started. This is a very long subroutine, and hence computationally expensive to compile, due once again to the interactive nature of the algorithm and the extensive error checking. It prompts the user for the specified information, does some explicit error checking automatically, and if those constraints are met, will echo the data to the user to make sure the correct values have been read. As the user progresses through the algorithm, the stored values are periodically output to a permanent data file that can be used repeatedly by making the appropriate choice in the DATAIN subroutine prompt. It was decided to

develop such an interactive package because of experience the author has in using software packages that require the development of an input data file. It is all too easy to make a single mistake in building such files, e.g. typing in a comma rather than a period, which causes the program to produce totally unreasonable results. Hopefully with the inclusion of the interactive subroutine such trivial errors will be kept to a minimum, and the user's lead-in time will be significantly reduced. In fact, with the prompting that is provided by the subroutine the user can simply work through the program in order to determine what parameters are required. For all the unit models in the simulator, default values are provided, an option that is included for those situations in which the user isn't able to provide his own model parameters. A word of caution here is that the simulator is not designed to model start up procedures of a grinding circuit. The interactive subroutine basically starts the simulator from zero initial conditions, i.e. the sump, mills and pipes are initially empty. As such it can take up to 400 minutes of simulated time in order for the circuit to achieve a true steady state, i.e. the solids mass flow rate out of the circuit equals the solids mass flow rate into the circuit, although after 100 minutes the output is close enough to a steady state for that data to be used as the initial conditions for subsequent runs. However, the numerical values produced by the simulator during this period should not be construed as being indicative of what

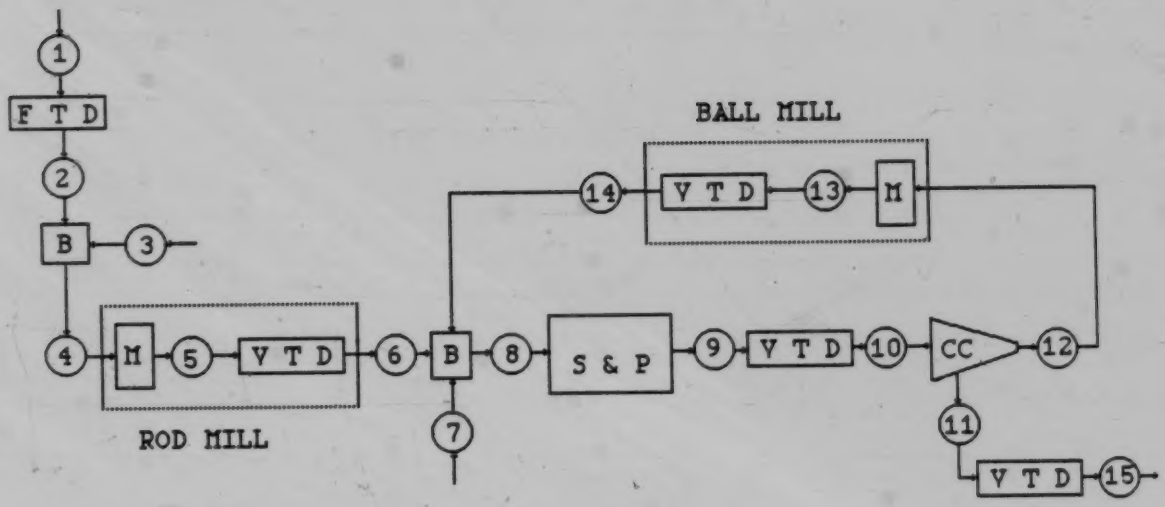
happens in an industrial grinding circuit during a start-up period. Therefore it is suggested that the simulator be run in an open loop when started from either the interactive subroutine or from the data file generated by the interactive subroutine.

Regardless of the initial option chosen, control is eventually returned to the driver which then proceeds to call the INIT subroutine which initializes the simulation parameters. A list of the default values are provided to the user concerning such items as the length of the simulation run, the incremental step size, etc. Also included in this subroutine are all the possible control options, with the default option being the open-loop simulation. If the user accepts all these values control is returned immediately to the driver. If the user doesn't accept these values, a menu is provided to allow any of the default values to be changed. Once again this an iterative process and the user can make as many or as few changes as required. Before control is returned to the driver a complete listing of the initialization values is dumped to the screen for final verification. Acceptance of the listed values returns control to the driver, otherwise the menu is presented again and the user can make further changes. As in the CHANGE subroutine, the menu format allows individual users the option of expanding the menu to fit their particular requirements. Once control has been returned to the driver from the INIT subroutine the work of the front end of the package has been

completed. All relevant data has now been loaded into the simulator and the only remaining prompting will be from the disturbance algorithm, as will be discussed shortly, and from the back end of the simulator that controls the output of simulator results to permanent data files for subsequent evaluation.

The middle section of the simulator is known as the integration section and it is this section that performs the actual dynamic simulation. It is here that some of the more important DYFLO subroutines are used which have been designed to numerically solve the differential equations in the simulation and to control the progress of the simulator according to the independent variable, in this case time. The DYFLO utilities are discussed in greater length in a subsequent section. An important comment to make here though is that one of the DYFLO subroutines controls the dumping of time dependent data to an output file during the simulation, one of three output devices available to the user. At every incremental step through the simulation, a call is made to the DISTRB subroutine to allow the user options for introducing various disturbances to the circuit. A flag can be set in the INIT subroutine to indicate that the simulator is to be run with no disturbances, or if disturbances are to be included the user can specify the time of the onset of the disturbance. When the simulator reaches the time of the onset of the disturbance the user is prompted with a menu of available disturbances, e.g. change in feed flow rate,

change in feed composition, extraneous water, feed size distribution etc. Once the user chooses a type of disturbance the program prompts for the nature of the disturbance introduction, e.g. step change, ramp change, square waves, multiple ramps, etc. Various flags are used to control the calling of this algorithm throughout the simulation, for example, for a single step change the subroutine is only called once, whereas for multiple ramp changes it will be called at every incremental time step. This is another of the menu driven sub-programs so it is quite easy to expand any part of the algorithm to accommodate individual user specified disturbances. Apart from the calls to the DYFLO and DISTRB subroutines, the integration section involves nothing more than a call to the CONFIG subroutine at each incremental time step. The CONFIG subroutine orders the calls to the specific unit model subroutines that comprise the given circuit. A simple integer flag is used to indicate which grinding circuit flowsheet is being simulated, and the call statements are simply ordered according to the relative placement of the units in the flowsheet. Currently only the three basic circuits are available, so the circuit flag ICIRC assumes a value between 1 and 3, see Figures 3.9 to 3.11 for schematic diagrams of the simulation flowsheets currently available. Obviously a user dealing with a circuit different from those already available simply expands CONFIG to include the appropriate series of calls to the unit models. Since CONFIG is called by the driver at every



MODEL LEGEND:

FTD = Fixed Time Delay
B = Blender
M = Mixer
VTD = Variable Time Delay
S&P = Sump and Pump
CC = Cyclone Classifier

INPUT STREAMS:

1 = Fresh Ore Feed
3 = Rod Mill Water
7 = Sump Water
9 = Sump Water

FLOW STREAM IDENTIFICATION:

→(N)→ Flow Stream Number "N"

Figure 3.9 Schematic of Reversed Closed Circuit Simulation Flowsheet

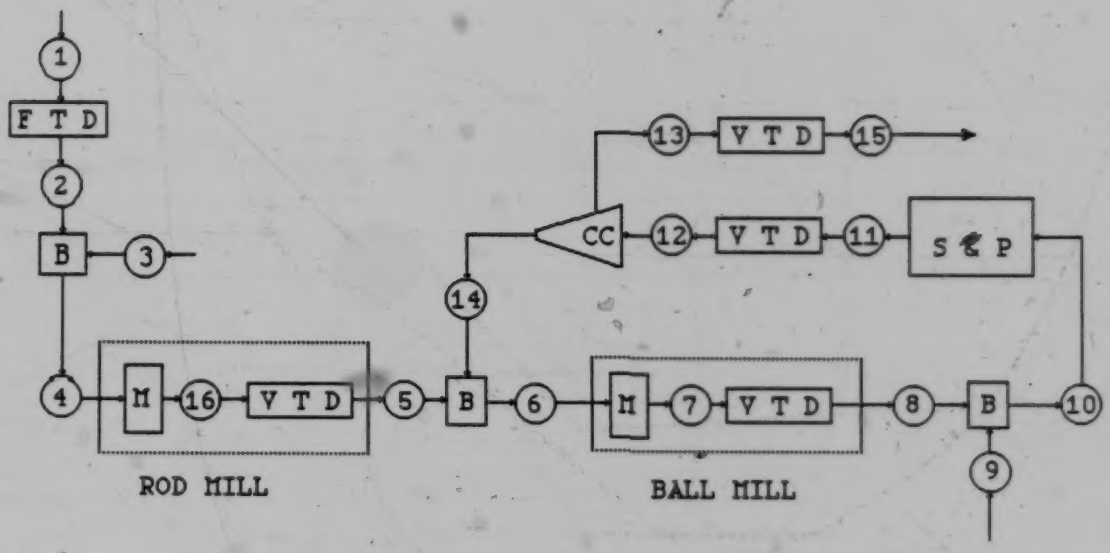


Figure 3.10 Schematic of Conventional Closed Circuit Simulation Flowsheet

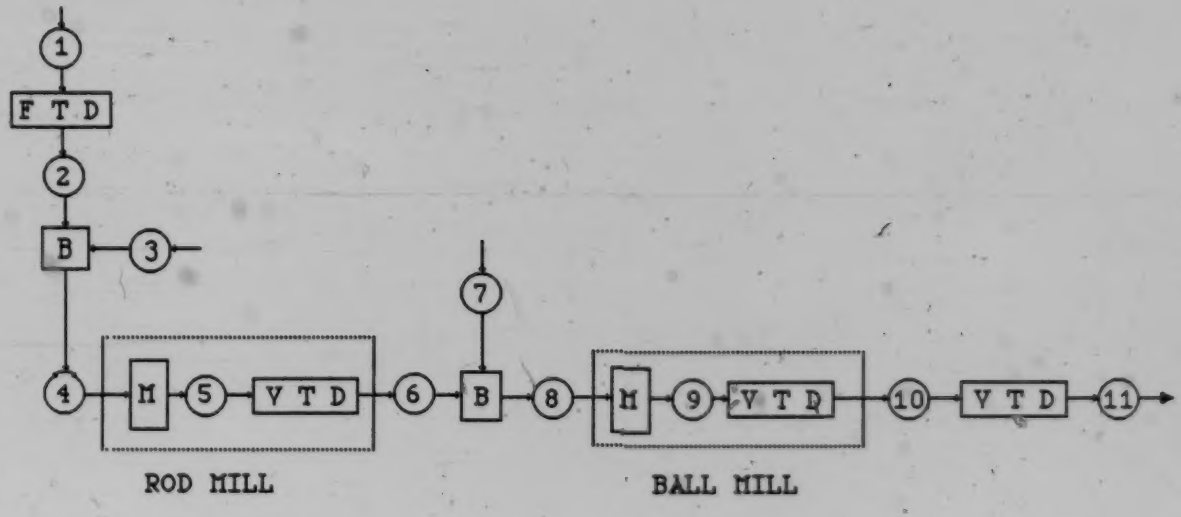


Figure 3.11 Schematic of Open Circuit Simulation Flowsheet

integration step it is quite easy to append various tests or calculations in this subroutine in order to output information specific to a particular run. Since CONFIG is quite small it can be altered and compiled quite easily.

Once the DYFLO subroutines have stopped the simulation at the specified point in time the driver performs only one more function, and that is a call to the OUT subroutine. This provides the user with two more options of retaining the simulator results in permanent files, the first being the time dependent printout produced while the simulator executes. The second file, if requested, will contain a tabular record of the rod and ball mill breakage and selection functions and a complete synopsis of the micro and macro properties of the important streams in the circuit. If this option is required, two subsidiary subroutines MILOUT for the breakage and selection functions, and STROUT for the stream synopsis are called from the OUT subroutine. Due to page limitations this synopsis is effectively constrained to fifteen streams, which for some simulations entails making a choice of which streams to include. The third output file can be used to dump the entire simulator data to a permanent file so that the simulator can be rerun using this data as the starting point. The use of the OPEN command in the OUT subroutine allows the user to output data to previously established files, with the understanding of course that any data already in these files will probably be lost. A second possibility is to declare a file name as NEW, and the data

will be written to a new file that didn't exist prior to the simulator being run. A word of caution here is to keep a good record of what files exist on which disk or account because declaring an existing file as NEW or a non-existant file as OLD will result in a fatal execution error that usually requires rerunning the simulator. Once control is returned to the driver from the OUT subroutine the simulation has finished.

3.3.2 COMMON BLOCKS

As indicated earlier, COMMON blocks are used in the simulator instead of subroutine argument strings to pass large blocks of variables between subprograms. There are currently three common blocks in use in the simulator, three of which are fairly concise and require little explanation, but the fourth contains all the information specific to the circuit being simulated and due to its size and complexity requires some discussion. The four common blocks currently in use are DATA1, CINT, CHAR1 and CNTRL. CINT carries the information relevant to the DYFLO subroutines such as the value of the independent variable, the incremental step size, the order of integration, etc. It has been well documented by Franks (1972) and used in the simulator without alteration. CHAR1 currently holds only the character string TITLE which is a CHARACTER*72 variable used to carry the simulation title. CNTRL holds the control parameters that are required for the closed loop simulations as will be discussed in the next

chapter. The most important common block is the DATA1 block, which has the following form, i.e. in FORTRAN source code.

```
  COMMON /DATA1/ ICIRC, NCOMP, NSTRM, NSIZE, NOMILL,
1               SUMP(45,2), RMILL(40,6,2), PLUG(44,20,5),
2               DATA(22,2), STREAM(44,20), CYCLON(12)
```

The first five variables are integer variables that are used at various points in the simulator to control program flow. Briefly they are as follows:

1. ICIRC indicates the type of circuit configuration to be simulated, currently ranges from 1 to 3, with ICIRC=1 for the reversed closed circuit, (pre-classification), ICIRC=2 for the standard closed circuit (post-classification), and ICIRC=3 for the open circuit grinding simulator.
2. NCOMP indicates the number of solid components in the feed stream, currently the maximum is 2, with a note here that expansion to 3 or more components, although not that difficult a problem given a good understanding of how the simulator works, is certainly a non-trivial and time consuming project.
3. NSTRM indicates the number of streams being monitored through the simulation. Although of great importance for diagnostics in the developmental stages its major use now is in controlling the simulator output.
4. NSIZE is used to indicate the number of size classes

being monitored by the simulator and is used extensively in the iterative process of calculating size class data in the unit modules. Currently limited to a maximum of 20 which is considered to be the practical limit of the requirements for this type of work.

5. NOMILL indicates the number of mills in the simulator. Although all circuits currently use two mills, future expansion may involve the development of single or multiple mill simulators in which case the value of NOMILL becomes significant.

The remaining variables are 1, 2 or 3 dimensional arrays that carry information specific to a particular unit model or relevant to the streams in the circuit.

1. SUMP(45,2) retains important information regarding the sump/pump models used in the simulator. In column 1, elements 1 and 2 contain the sump dimensions, while 3 is a flag used to indicate unusually high or low pulp levels in the sump. Cell 10 indicates which type of pump model is required, currently ranging from 0 to 3, while cells 14-15, 24-26 and 34-38 inclusive are used as pump parameters for the chosen model. Column 2 uses the first 40 cells to hold information about the individual size classes in the sump pulp, while 41-45 contain the solids mass holdup, water mass holdup, volume holdup, % solids in the slurry and the sump level data respectively. As

can be seen there is some space in the first column for additonal information should the user require it.

2. RMILL(40,6,2) is a large array that contains all the information needed by the mill models. The third dimension is set to 1 for the rod mill and 2 for the ball mill so the call to the MILL subroutine simply has to specify the mill number. Of the six columns allocated to each mill the first two contain the breakage and selection functions for component 1 and 2 respectively. If a single component feed is being simulated the second column will contain all zeroes. Since the mill model can account for up to three perfect mixers in series, columns 3, 4 and 5 retain the mass fraction of component 1 (rows 1-20), and component 2 (rows 21-40) in each size class respectively. Column 6 contains other pertinent information such as the mill volume, # of mixers, individual mixer volumes and the mass fraction of component 1 in each of the three mixers. Once again there is excess space in this array for future expansion.
3. PLUG(44,20,5) contains the stream information as it is held up in the PIPE and CONVEY subroutines that simulate the variable and fixed time delays. The third dimension is set at 5 because this currently represents the maximum number of calls to these subroutines in any given circuit configuration. The second dimension indicates the maximum number of cells currently available in any given buffer. The 44 rows are used to store the 40 size fractions (i.e.

2·20 per component) and the four stream macro properties.

4. DATA(22,2) has been treated as a general catch-all for pertinent information that doesn't really belong in any other array. It was initially used to store the characteristic size classes only, with column 1, rows 1-20 still being used for this purpose. Row elements 21 and 22 in columns 1 and 2 are used to store the solids and water densities for the two components, and although the water density is independent of the solids density this value has to be included for accounting purposes in certain parts of the simulator. Column 2 is arbitrarily filled with other pertinent data that will often be different for different circuits. For example, DATA(19,2) is used to specify the pulp density in the rod mill, in the conventional closed circuit DATA(20,2) specifies the steady state flow rate to the cyclones given that this configuration currently runs with a fixed speed pump, whereas in the reversed closed circuit DATA(20,2) specifies the steady state flow rate of water into the sump. The first ten cells of column 2 are reserved for the various flags involved in the disturbance subroutine. Although this is not a very suitable arrangement it makes practical use of the available storage space, but care should be taken to avoid confusion in this array as to which variables are located in a particular position.
5. STREAM(44,20) contains the micro and macro properties of each stream within the circuit. The 44 rows contain the

40 size classes and the 4 macro properties, solids mass flow rate, water mass flow rate, volumetric flow rate and per cent solids. The twenty columns identify the each individual stream, so that for a call to any unit subroutine only the column number has to be specified. Although not under current consideration, it is obviously quite easy to expand the number of columns in the STREAM array in order to handle larger circuits.

6. CYCLON(12) simply contains the 12 cyclone parameters required by the simulator. These are the model parameters as outlined in equations 3.10 to 3.13. If another cyclone model is used this array can be expanded to handle such requirements.

3.3.3 DYFLO Utilities

The work of Franks (1972) has been referenced repeatedly in this work, so it is appropriate at this time to make a few comments regarding the DYFLO subroutines used in the simulator. The original work of Franks was published at a time, when the true value of computer simulation was first being realized. The object of this work was to provide a systematic framework within which individuals could build their simulation software. This in turn was projected to reduce the effort required to design the simulator itself, and enable the engineer to devote more time to the important area of analysing the results of the simulations and the correlation of those results to the real process in order to improve

upon the overall performance of that process. Two major points are stressed by Franks, the first being one of building the simulator in such a way that the mathematical models are arranged in a logical or cause-and-effect sequence, the second point being that a modular structure not only accurately simulates a real process, but increases the flexibility of the simulator and aids in the design and development stage. This philosophy has been carried over to the design of the grinding circuit simulator.

Three important DYFLO subroutines used in the simulator are INTI, INTG and PRNTF. INTI controls the intregration operations and steps the independent variable (time). A call is made to INTI at each integration interval, which requires the passing of three arguments, the independent variable, the integration increment and the flag indicating the order of integration required. As indicated earlier, due to the stiff system of differential equations encountered in the MILL subroutine, the order of integration has been defaulted to first order integration. As such, INTI simply increments time and the simulation continues. For higher order integrations, INTI provides the mechanics to permit multiple passes through a given time interval. INTG does the actual integration of the independent variables and requires just two arguments, the name of the integrated variable and the derivative of that variable. It can perform first, second and fourth order integration based on the value of the flag provided by INTI. Given that X is the integrated variable,

and DX is the derivative, then the first order solution is of the form

$$X_{i+1} = X_i + DX_i \cdot DT \qquad (3.28)$$

where DT is the integration step size. This subroutine is called from various unit models in the simulator that require the solution of a first order differential equation. A third major DYFLO utility to be discussed here is the PRNTF subroutine which ouputs user specified time dependent variables at specified time intervals and also tests for the end of the simulation. The author has made slight modifications to this algorithm by including a section to output the relevant column headings for these variables, which are printed out in the first call to PRNTF algorithm. This has been done in an effort to make the subroutine all inclusive and would require some alteration to the source code if a different set of time dependent variables were required. Due to the limitations in paper width the maximum number of variables that can be printed out is about ten. If more values are required the user has to decide if another file is required or if some the variables currently included can be deleted.

DYFLO also provides the utilities for implementing proportional (P), proportional integral (PI) and proportional integral derivative (PID) feedback control. Due to the nature of the system at hand only the P and PI controller

modes are currently available in the simulator. These algorithms are completely transparent to the user, requiring only specification of the appropriate controller gains and set-points for the output variables. Although Franks developed the control algorithms using the proportional band (PB) and reset rate (RPT) to relate the controller output to the error signal, it is more common today to use the proportional gain K_c and the integral time constant τ_I, where

$$K_c = \frac{100}{PB} \quad \text{and} \quad \tau_I = \frac{1}{RPT}$$

To avoid confusion the user is prompted for the proportional gain and integral time constants only, with the appropriate conversions being made within the calling program. A more detailed discussion of the contol implementation process is reserved until the next chapter.

3.4 Open Loop Simulations

At this stage, it is hoped that the reader has a fairly good idea as to how the simulator operates. The next task is to run the simulator under various open-loop conditions and monitor the output. In order to tie together the previous comments, the results of an interactive session showing the various options that are available to the user has been included in Appendix A. Once again, the reader should keep in mind that it is only the reversed closed circuit that is being considered in this discussion. Also included is an

example of the tabulated output of the time dependent variables used to generate the subsequent plots and a listing of the stream synopsis available at the end of each simulation run. Not included for obvious reasons is a listing of the input data file which really has very little reference unless the reader is familiar with the data input and output code.

An additional point to be made is that the circuit being modelled is relatively small by some standards, 64 tonnes/hr, and as such responds fairly fast to disturbances, attaining a new steady state usually within an hour from the onset of the disturbance. Unfortunately, the simulator output cannot be verified directly with industrial data because, on the whole, such data are simply not available. However, from a qualitative standpoint, the simulator should be able to produce definite trends in the output that, based on the users experience or knowledge of the underlying mechanics of the circuit, behave as expected. For example, if the hardness of the feed ore increased, one would expect to see the product stream becoming coarser and an increase in the density of the cyclone feed stream. These two factors are simply a reflection of the increase in energy requirements for producing a product stream with a specified size distribution. For all the simulations included in this section the variable speed pump is being manipulated in order to control the sump level so as to avoid sump overflow or sump empty conditions. Discussion of the design of this

controller is left until the next chapter, suffice to say at this point that a proportional controller with a gain of 2 is used to impart fairly tight control over the sump level.

In running open-loop simulations, it is necessary to seriously evaluate the possible disturbances that could affect the grinding circuit. The two major input streams, fresh ore feed and sump water, are fairly obvious candidates as disturbances. The rod mill water is not included as it is assumed to be ratioed to the fresh ore feed rate. The hardness and size distribution of the feed stream and the volumetric flow rate of water to the sump can all affect the circuit performance. The compounding problem with these disturbances is that they cannot be measured directly and as such any potential control action can only be taken once the effect is felt in the response variables, i.e. feedback control. Other disturbances relate to the machinery of the circuit such as mill liner wear, cyclone liner wear and pump impeller wear. These factors take a much slower toll on the circuit performance, and are usually imperceptible on a day to day basis, but can be a significant detriment over the long run. As yet such considerations have not been included in the simulator and it is the authors contention that if a good feedback control system is implemented to handle the major disturbances then it will also be able to handle these long term, slow moving disturbances. More acute circuit upsets can be caused by equipment failure, probably the most common being pump failure. Once again this is not included

in the simulator as it is felt that such problems are better handled at the operations level, i.e. provision for backup equipment and the appropriate switching mechanisms so that the backup equipment is switched on the moment the initial equipment fails.

The menu driven disturbance subroutine described earlier, DISTRB, provides options for four different types of disturbances,

1. Change in feed stream components (used to model hardness changes)
2. Change in feed stream size distribution
3. Change in feed stream mass flow rate
4. Change in water flow rate to the sump

Once again, individual users can expand this menu as they see fit, but it is the authors contention that given the simulator's current limitations these are the only four viable disturbances available. For example, one may wish to investigate the effect of pulp density in the rod mill on grinding performance. In order to do this, the dependence of the breakage and selection functions on pulp density would have to be investigated, and these results included in the simulator mill models. The third option above is included in order to generate the process reaction curves required for controller tuning, and as such is not considered a true disturbance. The feed stream size distribution will be shown to

have a negligible effect on circuit performance and is included for reasons of completeness. Although all four disturbances will be examined to complete this chapter, it is necessary to indicate that it is only the feed stream hardness changes and the extraneous sump water additions that will be considered in detail in future control work.

3.4.1 Increase in Fresh Ore Feed Hardness

In an industrial situation it is virtually impossible to test the effects of a change in the ore hardness on the dynamics of the grinding circuit. However, given a valid simulator, such disturbances can be studied and evaluated in detail at very little cost. The hardness change is simulated by making a step change from component 1, the soft ore, to component 2, the hard ore. The two ores have identical breakage elements, but the selection elements of component 2 are 50% of the corresponding elements of component 1. This corresponds to approximately a 30% increase in the work index of the feed ore. Figure 3.12 shows the effects of an increase in ore hardness introduced at two minutes into the simulation. As expected there is a significant lag period until the effects are felt in the circuit variables monitored. The initial effect of this disturbance is to coarsen the rod mill discharge, which in turn, after a delay due to the sump, increases the mass fraction of material reporting to the cyclone underflow. The circulating load, measured as the ball mill discharge mass flow rate, starts to increase

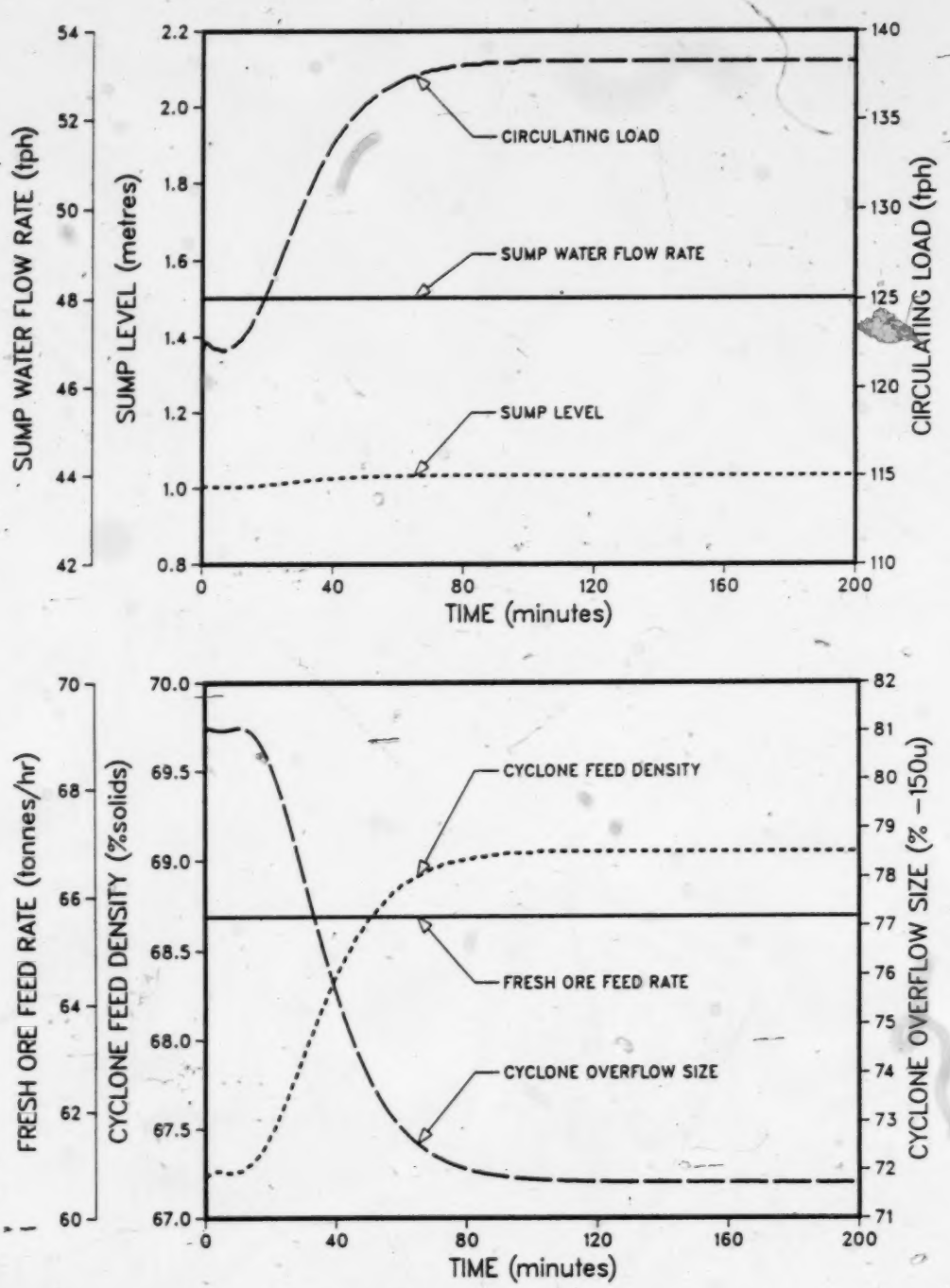


Figure 3.12 Open-Loop Response to an Increase in Ore Hardness at t=2 Minutes

only after a further delay due to the holdup in the ball mill. As the total volumetric flow rate to the sump increases, the pump speed increases in an effort to hold the pulp level close to the setpoint. The overall effect then is to pump a higher density feed to the cyclone at a higher volumetric flow rate. Since the dynamics of the cyclone have been ignored the cyclone feed density and the cyclone overflow size (% -150μ) react simultaneously, with both effects lagging slightly behind the response of the circulating load. The increased volumetric flow rate to the cyclone tends to reduce the cyclone cut size, d_{50}, but this is more than offset by the increased density of the cyclone feed, which acts to increase d_{50}. The overall effect then is to produce a coarser product stream, with the final steady state value being almost 10% below the desired size distribution of 81% -150μ.

Given that one of the potential operating goals is to produce a product stream with a specified size distribution, and that this size distribution results in optimal recovery in the downstream process, the occurrence of a hardness change in an uncontrolled circuit can be seen to be detrimental to the overall plant performance. In the next chapter control algorithms will be explored that will attempt to minimize the effects of ore hardness changes. In comparison to the disturbances discussed below, the circuit takes quite a long time to settle to a new steady state, almost 2 hours in the case of the cyclone overflow size distribution. This

is indicative of the holdup within the circuit, and the amount of time required to remove the material in the circuit at the time of the disturbance onset.

3.4.2 Increase in Sump Water Flow Rate

The flow rate of water to the sump can be manipulated in order to control various circuit parameters, or in certain situations can act as a disturbance to the circuit. Thus in essence one can consider having two separate streams flowing into the sump, the first being the manipulated stream, the second acting as the extraneous water that causes the disturbance. The simulator handles this problem by blending the two streams, using the WATBLD subroutine, and then monitoring the blended stream as it discharges into the sump. Thus this particular open-loop simulation serves two purposes in that it indicates the effect of extraneous sump water on circuit performance, and at the same time produces the data required for the process reaction curve required in the controller tuning exercise. Figure 3.13 represents the simulator response to a 20% increase in the sump water flow rate. Due to the sump acting as a perfect mixer, and the lack of dynamics around the cyclone, there is an almost immediate response in the cyclone feed density and the cyclone overflow size distribution. The increase in the volumetric flow rate to the sump acts to increase the sump level which is compensated for by an increase in the pump speed which in turn increases the volumetric flow rate to

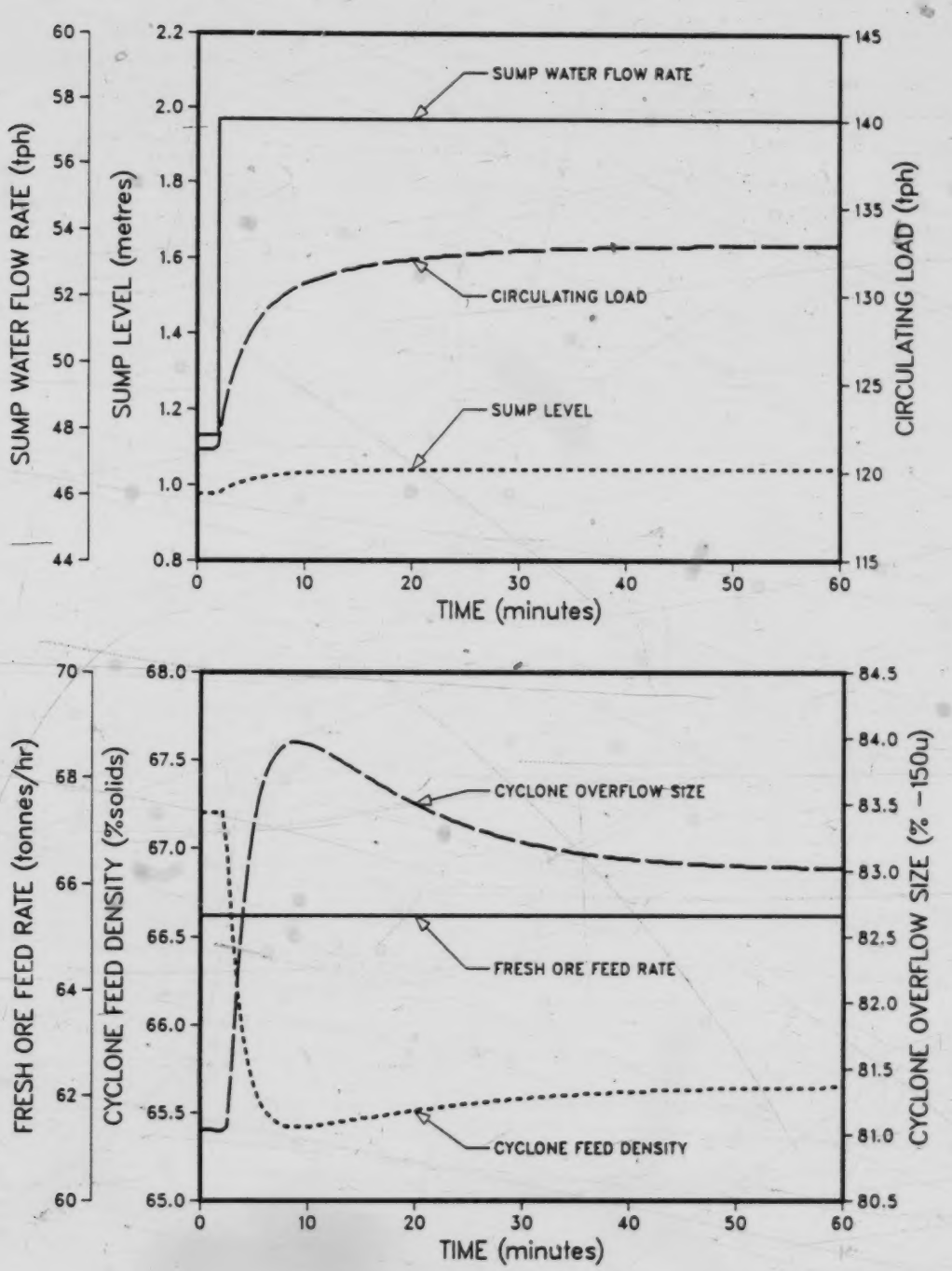


Figure 3.13 Open-Loop Response to a 20% Increase in Sump Water Flow Rate at t=2 Minutes

the cyclone. Since the VTD models are true pipes in this case, the effect is manifested immediately at the cyclone. The first material into the cyclone at this increased flow rate is that which comprises the pipe holdup volume, although there is no change in other operating variables, the increased flow rate acts immediately to decrease d_{50} and increase the fineness of the products stream. d_{50} is further reduced as the less dense slurry from the sump hits the cyclone, resulting in a rapid increase in the fineness of the product stream within a few minutes of the disturbance onset. As d_{50} is reduced, there is an increased routing of material to the underflow corresponding to an increase in the circulating load. This response is slightly delayed due to the holdup in the ball mill, but once this material returns to the sump, the cyclone feed density increases slightly resulting in an increase in d_{50} and a coarsening of the product stream. As Plitt (1976) has indicated, the cyclone feed density has a more pronounced effect on d_{50} than does the volumetric flow rate, but in this situation they both combine to move the classifying action in the same direction. After about 60 minutes, the circuit achieves a new steady state, with a higher circulating load, a reduced feed density and a finer product stream than was apparent before the increase in sump water flow rate. The product stream goes from 81% -150μ to 83% -150μ, which in itself may not be significant, but may represent a loss of efficiency due to overgrinding of the ore. Because the effect of the increased

recirculating load on the product stream size distribution is opposite to that of the reduced density of the cyclone feed stream, second order effects are apparent in the trajectory of the output variable. This behaviour is important to note because of the problems that will be encountered when it comes time to design the controller for this situation.

3.4.3 Increase in Fresh Ore Feed Rate

As indicated earlier, the fresh ore feed rate in itself is not considered as a disturbance, but rather a manipulable variable. As such, knowledge of the response of the circuit to a change in the feed rate is important. Figure 3.14 shows the circuit response to a 20% increase in the fresh ore feed rate introduced at two minutes into the simulation. The transients of the monitored variables are quite similar to those seen in the hardness increase simulation except they are realized much sooner. This is due in the most part to the change in volumetric flow rate that accompanies an increase in the mass flow rate. Since the mills in the simulator are considered to be constant volumetric holdup overflow discharge mills, the effects of a change in volumetric flow rate is transmitted around the circuit faster than, say, a change in the ore hardness. The modification to the VTD algorithm to account for this type of behaviour, as discussed in section 3.2.4, can be shown to delay the circuit response by one or two minutes, a delay that is felt to be a more accurate reflection of the real situtation. If the original pipe

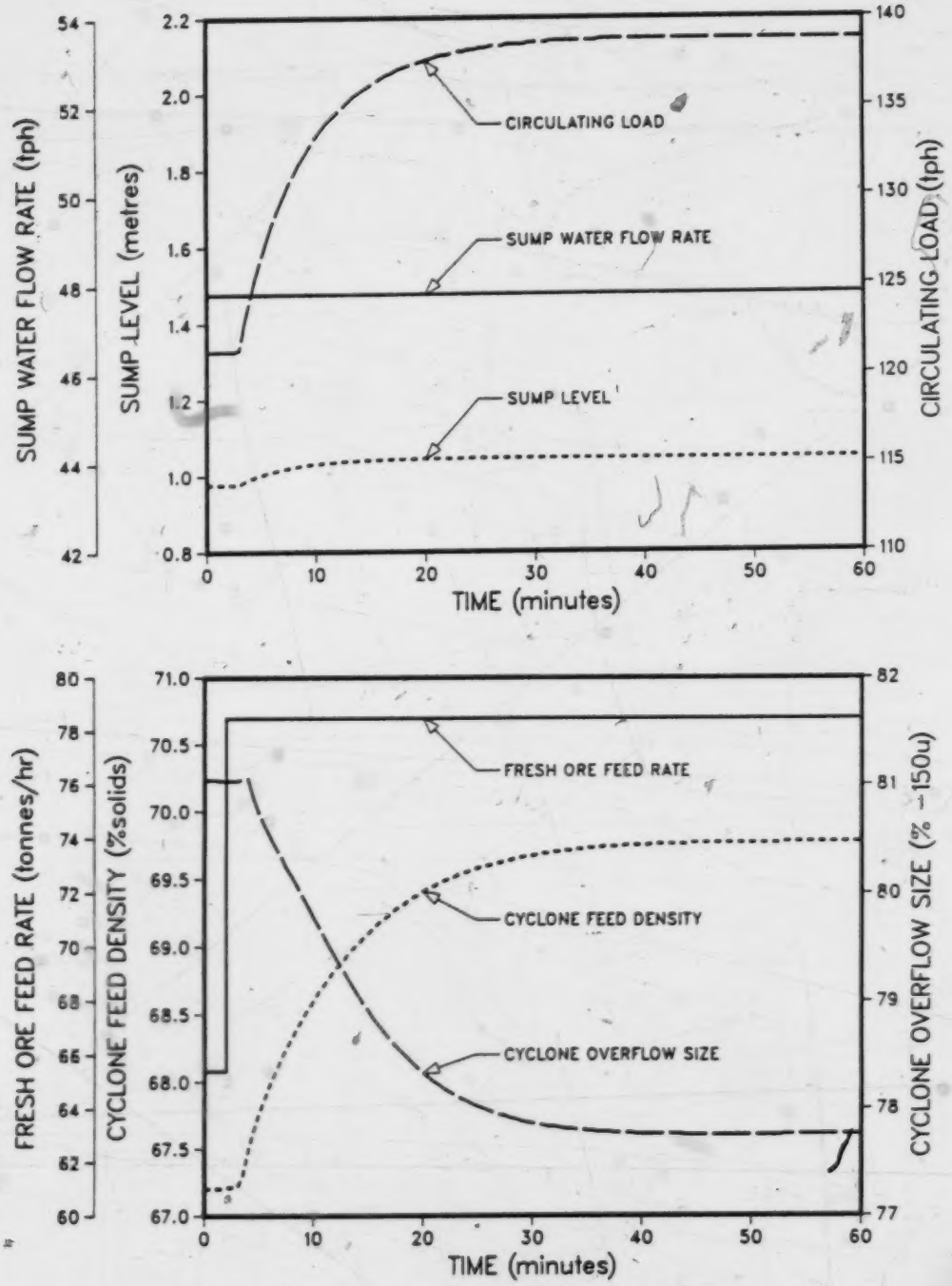


Figure 3.14 Open-Loop Response to a 20% Increase in the Fresh Ore Feed Rate at t=2 Minutes

models are included, the response around the circuit is almost immediate, which given the hold-up, or inertia in the system, is probably quite unrealistic. From the time of the disturbance initiation, the response of the circuit is delayed first by the conveyor, about 20 seconds, then by the rod mill, so that the cyclone feed density doesn't increase until almost two minutes later. Once again the pump speed increases to maintain the set-point sump level in response to an increased volumetric flow rate to the sump. Although these two factors tend to push d_{50} in opposite directions, the overall result is a coarser product stream, yet at the same time the flow rate of solids to the underflow stream increases. This, in turn, causes a delayed increase in the circulating load, as more material reports to the underflow, and then an additional increase to the volumetric flow rate to the sump, with this process iterating until a new steady state is achieved. The net effect of an increase in the feed rate is a drop in the product stream size distribution from the specified 81% -150μ to 77.8% -150μ. The transients of the output variables exhibit a typical first order with time delay response, since the recirculating load in this case moves the response variables in the same direction as the initial disturbance. This is in contrast to the previous case in which the effect of the recirculating load was to move the response variable in the opposite direction to the disturbance.

3.4.4 Change in Feed Size Distribution

In certain situations it is recognized that the size distribution of the crusher plant product can vary. For example, if the plant is run while the tertiary crushers are shut down and all the material is routed through the secondary crushers there can be a slight, but noticeable increase in the product coarseness. As a result, it is important to monitor the effects of a change in the grinding circuit feed size distribution. As expected, Figure 3.15 indicates that an increase in the feed coarseness does affect the response variables, but certainly not to the extent of the other disturbances examined in this section. Similar to those trajectories seen in the hardness change simulation, there is a significant time delay while the coarser ore moves through the circuit. The rod mill discharge becomes slightly coarser which eventually initiates a coarsening of the product stream once the material moves through the sump. With an increasing fraction of the cyclone feed reporting to the underflow, the circulating load eventually increases, which in turn increases the density of the cyclone feed material. This process is similar to those described for the increase in ore hardness and increase in ore feed rate, with the response variables showing the typical first order with time delay trajectory. The simulator currently offers only two options for implementing a feed size disturbance, an increase and a decrease in the coarseness. To increase the coarseness, the finer two size classes are set to zero and a

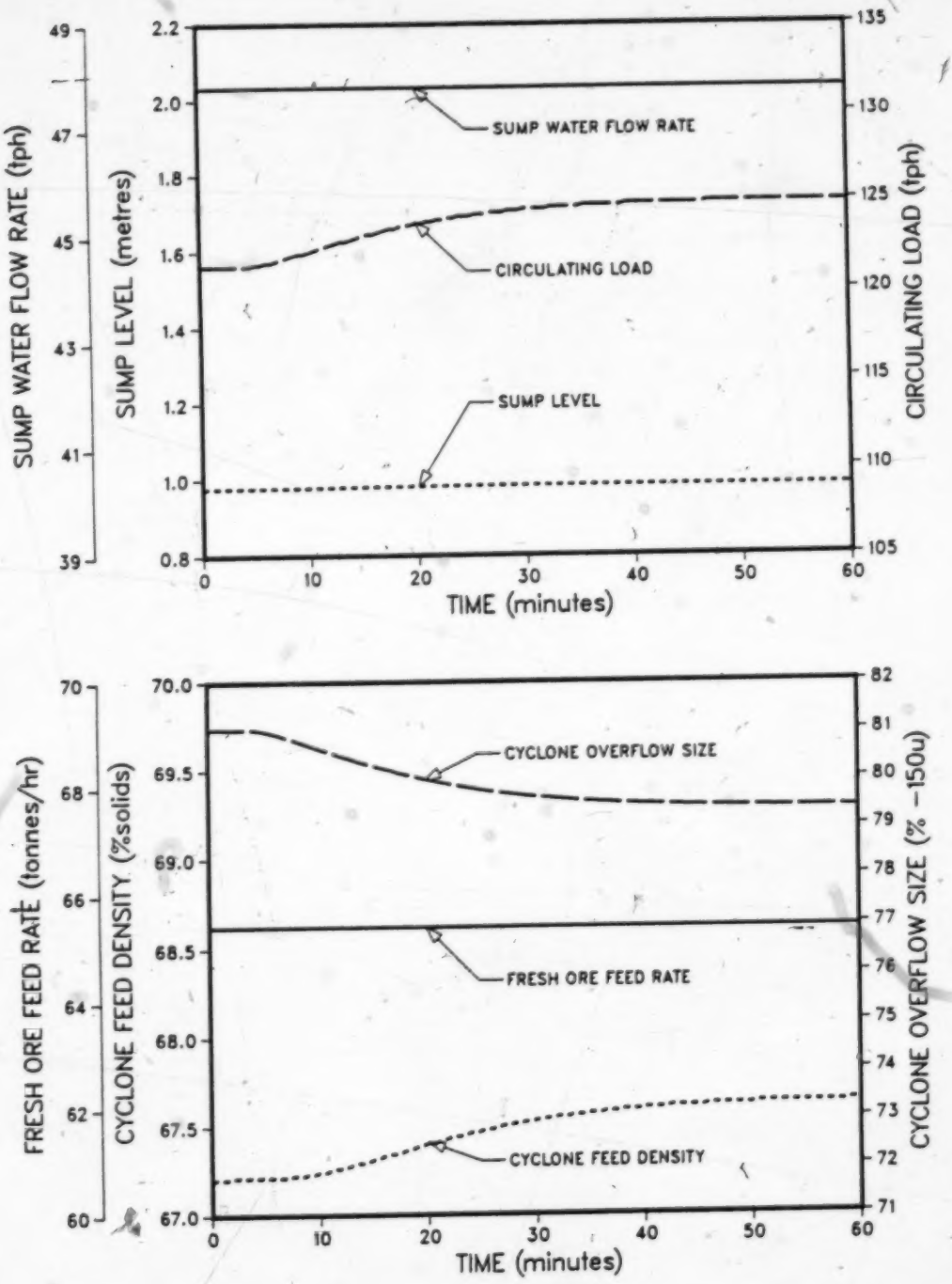


Figure 3.15 Open-Loop Response to an Change in the Fresh Ore Feed Stream Size Distribution

suitable multiplier chosen so as to redistribute this material throughout the remaining size classes. Although this is not that realistic, it is an efficient and effective method of simulating a change in the size distribution. To decrease the coarseness, a multiplier less than unity is chosen so that the mass fraction in each size class is reduced and the residual material is transferred to the pan fraction. These two algorithms have been chosen since they provide the option for manipulating the feed stream size distribution independent of the original size distribution. Once again, the individual user can make any alterations to the disturbance algorithms as they see fit, given the structure of the DISTRB subroutine it is possible to design an infinite variety of disturbance sequences to fit the users requirements.

4. Classical Feedback Control of the Grinding Circuit Dynamic Simulator

If one can accept the simulator performance on a qualitative basis as being representative of the true process then the next logical step is to study methods of controlling the circuit in the presence of various disturbances. Based on previous arguments, the main area of interest is regulatory control, in which system inputs are manipulated such that pre-specified performance criteria, or set-points are met. Servo control, in which the set-points are adjusted on-line, usually in response to changing economic or productiion constraints, is at this time considered to be of secondary importance.

In order to design a grinding circuit controller it is necessary to identify the operating goals. These can vary between installations depending on the type of ore being processed and in part on the management philosophy at that particular plant. However, the general aim is to produce an optimal grind at a specified tonnage for the lowest unit cost. In those operations dealing with low grade ores the goal could be one of maximizing tonnage while maintaining the grind. Other situations may call for the product size to be minimized for a constant throughput, e.g. the mill throughput is tied to the mine production rate with little or no buffer capacity between the two operations. Of course there are trade-offs between objectives, i.e. one or more cannot be sacrificed completely for the sake of others.

Where this balance occurs is a function of the plant economics and is usually plant or site specific. Regardless of what the operational goals are, for any given circuit there are performance constraints on each unit in the operation. These include the capacity constraint on the rod mill to avoid rod tangles, the cyclone constraint to avoid roping in the underflow and the related slurry density constraints on the ball mill. Such constraints have not as yet been included in the simulator, which implies that any results obtained through these simulations should only serve as qualitative approximations, and not be construed as the definitive solution of the grinding circuit control problems.

Once the operating parameters have been established it is necessary to define the nature of the control algorithm to be used. Classical control techniques involve treating the process as a single input single output (SISO) system such that one controlled variable is linked to one manipulated variable. The value of the system output (controlled variable) is compared to the set-point, with this difference being known as the error term $e(t)$, such that:

$$e(t) = y_{SP}(t) - \hat{y}(t) \tag{4.1}$$

where $\hat{y}(t)$ is the measured or estimated value of the output variable at time t. The controller operates on this error signal in such a way as to return the output to the desired value. Included in this system is some form of measuring

device to monitor the output and a final control element to affect changes on the circuit. The general form of the classical control algorithm is as follows:

$$c(t) = K_c \epsilon(t) + \frac{K_c}{\tau_I} \int \epsilon(t)dt + K_c \tau_D \frac{d\epsilon(t)}{dt} + c_s \qquad (4.2)$$

where:

$c(t)$ = the controller output signal
K_c = the proportional gain of the controller
$\epsilon(t)$ = the controller input signal, error term
τ_I = integral time constant or reset time
τ_D = derivative time constant or rate time
c_s = controller's bias signal or steady state offset

Equation 4.2 is generally know as the PID controller, or proportional integral derivative controller. In certain situations, a proportional only controller can be implemented, however, since the control action is proportional to the error term there is always some offset involved in the controlled variable. In those applications in which some offset in the controlled variable is not critical, such as pulp level in a sump, a proportional only controller may be the most suitable. Proportional-Integral (PI) control removes this offset since action is taken on the integral of the error term so that the controller output, i.e. the signal to the final control element, will continue to change as long as an error exists in the process output. The penalty for including integral control action is that the order of the

closed-loop system is increased which in turn slows down the response of the system. From the open-loop response curves presented in the previous chapter it is readily apparent that the grinding circuit acts as a fairly sluggish multi-capacity process. As a result, any PI control action will have to manipulate the inputs slowly in order to avoid stability problems. By including derivative control action, the PID controller anticipates what the error will be in the immediate future and applies control action proportional to the rate of change of the error term. Unfortunately, in most industrial applications there is a certain degree of noise associated with the input signal which can cause large swings in both the sign and magnitude of the derivative term resulting in poor or incorrect control action. Obviously there are ways to circumvent this problem such as signal processing and/or filtering, but for the purposes of this study only PI control will be considered.

In classical control strategies, each manipulated variable is linked to a controlled variable on a one-to-one basis to form the SISO system. Given an industrial grinding circuit that has multiple inputs and outputs, a MIMO system, it is necessary to link each output with an appropriate input in order to implement control. From the open-loop response curves in the previous chapter it is obvious that a change in any given input results in changes to more than one of the outputs, a problem known as system interaction, and one that is a well recognized limitation to good control

of MIMO systems using classical control theory.

Herbst and Rajamani (1979) have listed four variables that require some form of control in grinding circuits, (1) product size distribution, (2) circulating load, (3) sump level, and (4) percent solids in the mill. Until recently, due to the expense of measuring the product size distribution, some installations attempt to control the cyclone feed density as an inferential control of the product size distribution. This however, doesn't change the number of output variables that require controlling. In the same presentation, Herbst and Rajamani identify four input variables that can be manipulated in order to control the system outputs, (1) fresh ore feed rate, (2) fresh water feed rate, (3) sump water feed rate, and (4) cyclone feed pumping rate. Using data from pilot plant and industrial scale grinding circuits, these authors developed what is known as the process matrix to qualitatively relate the response of controlled variables to changes in the manipulated variables, see Figure 4.1. For this current work the order of the process matrix can be reduced from four to three because the fresh water feed rate is ratioed to the fresh ore feed rate resulting in a constant pulp density in the rod mill. Thus fresh water feed rate is removed as a potential manipulated variable, and as a consequence, the percent solids in the rod mill is removed as a potential control variable. A positive sign in the process matrix indicates that an increase in the manipulated variable will result in a corresponding

Manipulated Variable	Controlled Variable: Product Fineness	Circulating Load	Sump Level	Mill % Solids
Sump Water Addition Rate	+ Fast	+ Fast	+ Fast	+/- Slow
Fresh Feed Solids Rate	- Slow	+ Slow	+ Slow	+ Fast
Cyclone Feed Pumping Rate	+ Fast	+ Fast	- Fast	+ Fast
Feed Water Addition Rate	+/- Slow	+ Slow	+ Slow	- Fast

Figure 4.1 Process Matrix for a Grinding Circuit (after Herbst and Rajamani)

Controlled	Manipulated
Sump Level	(1) Pumping Rate or (2) Sump Water Addition Rate
Particle Size (or cyc. feed density)	(1) Fresh Feed Solids Rate or (2) Sump Water Addition Rate
Circulating Load (or mill throughput)	(1) Sump Water Addition Rate or (2) Fresh Feed Solids Rate
Mill % Solids (or feed % solids)	(1) Fresh Feed Water Rate

Figure 4.2 Linking of Controlled and Manipulated Variables for Classical Control Strategies (after Herbst and Rajamani)

increase in the controlled variable, whereas a negative sign indicates a decrease in the controlled variable. The speed of the response is simply indicated as fast or slow. Inspection of Figures 3.17 and 3.18 verifies the first two rows of the process matrix. The third row of the process matrix will be verified when the effect of pump speed on circuit performance is evaluated later in the chapter.

Herbst and Rajamani also include a listing of the most appropriate linking of manipulated variables with controlled variables, Figure 4.2. Once again, for the purposes of this work the linking of fresh water feed rate to mill % solids can be ignored. Since the simulator now includes an option to use a variable speed pump to control sump level, the system has in effect three inputs, fresh ore feed rate, pump speed and sump water flow rate and three outputs, circulating load, sump level and product size distribution, and as such is referred to as a 3x3 system. Given that the pump speed can be dedicated to controlling sump level, two distinct types of classical control strategies can be identified, with Herbst and Rajamani coining the terms Type I and Type II to describe them. Type I controls product size with fresh ore feed rate and circulating load by sump water addition rate, whereas Type II uses sump water addition rate to control product size and fresh ore feed rate to control the circulating load.

It is also necessary to examine the potential disturbances that affect the circuit performance. From work in the

previous chapter, from a regulatory control aspect, it is changes in ore hardness and size distribution and the extraneous flow of water to the sump that can be identified as potential circuit disturbances. Of these, it was shown that changes in the feed ore size distribution had only minimal effect on the circuit performance. However in certain situations when rod mill tonnage is being pushed close to capacity, the feed ore size distribution becomes a critical factor in avoiding rod tangles, Bradburn et al (1977). Once again, the user with such applications in mind is well advised to build in the specific constraints as required. Of the remaining two disturbances, it is felt that hardness changes provide the most serious control problem, and it is this area that will be investigated first.

The essence of this current chapter is to investigate controlling the cyclone overflow size distribution by manipulating the fresh ore feed rate given a step change in the ore hardness. While this may sound simple enough, it will become quite apparent that the system interactions, which until now have only been discussed in a qualitative manner, will play a major role in affecting the controller implementation.

4.1 Implementing Control on the Simulator

By default the simulator runs open-loop simulations. The user has the option to choose a closed-loop simulation via the menu provided in the INIT subroutine as described in

the previous chapter. Once a closed-loop simulation is indicated subroutine CONTRL is called to enable the user to choose the required control algorithm. Default values are provided for the set-point, and proportional and integral controller gains, the user has the option of accepting these values or of inputting other values. A separate COMMOM block, labelled CNTRL, has been created to carry the important control parameters. The most important of these is the integer flag ICNTRL which indicates the specified control algorithm. The actual control implementation is done in subroutine CIMPL which is called from the CONFIG subroutine at each integration step. Depending on the value of ICNTRL different algorithms in CIMPL are executed in order to implement control. Once again, due to the modular structure of the simulator and the menu driven execution, expansion of this part of the package is not difficult. To facilitate use of the interactive CONTRL subroutine, the user specifies only the positive values of K_c and τ_I, the algorithm automatically converts these values to the proportional band (PB) and repeats per minute (RPT) variables as required by the DYFLO control subroutines. Positive values only are required since the variable AXN is set according to whether direct (AXN=1) or reverse (AXN=-1) control action is required. Some checking is done of the user supplied set-point to ensure that unattainable or unrealistic set-points are not included in the simulator. Figure 4.2 shows schematically the structure of the closed-loop simulator, cf Figure

3.8. The DYFLO subroutine CONTR2 is called from CIMPL to perform the PI control calculations. Since the controller output is a function of the integral of the error signal, a call has to be made to the INTG subroutine in order to perform the integration. This call is made directly from the CIMPL subroutine in order to make the control algorithm as compact as possible. All calculations required to determine the controller input signal, and convert the normalized output signal to the appropriate value are performed in CIMPL. By combining all the necessary calculations in one subroutine each control module functions as a stand alone unit which enhances the flexibility of the overall system.

4.2 Design and Evaluation of Feedback Controllers

As indicated in the introductory comments, the control engineer has to decide which control algorithm to implement, P, PI, PID, etc., and how to link the various controlled and manipulated variables. Once this decision is made the next step is to choose the appropriate values of K_c, τ_I and τ_D and use some non-subjective evaluation of the controller performance in order to implement the best control on the process.

For classical control strategies various techniques have been proposed for controller tuning using the process reaction curve method which assumes that the process transfer function can be adequately described by a first order plus dead time model. These methods all involve determining

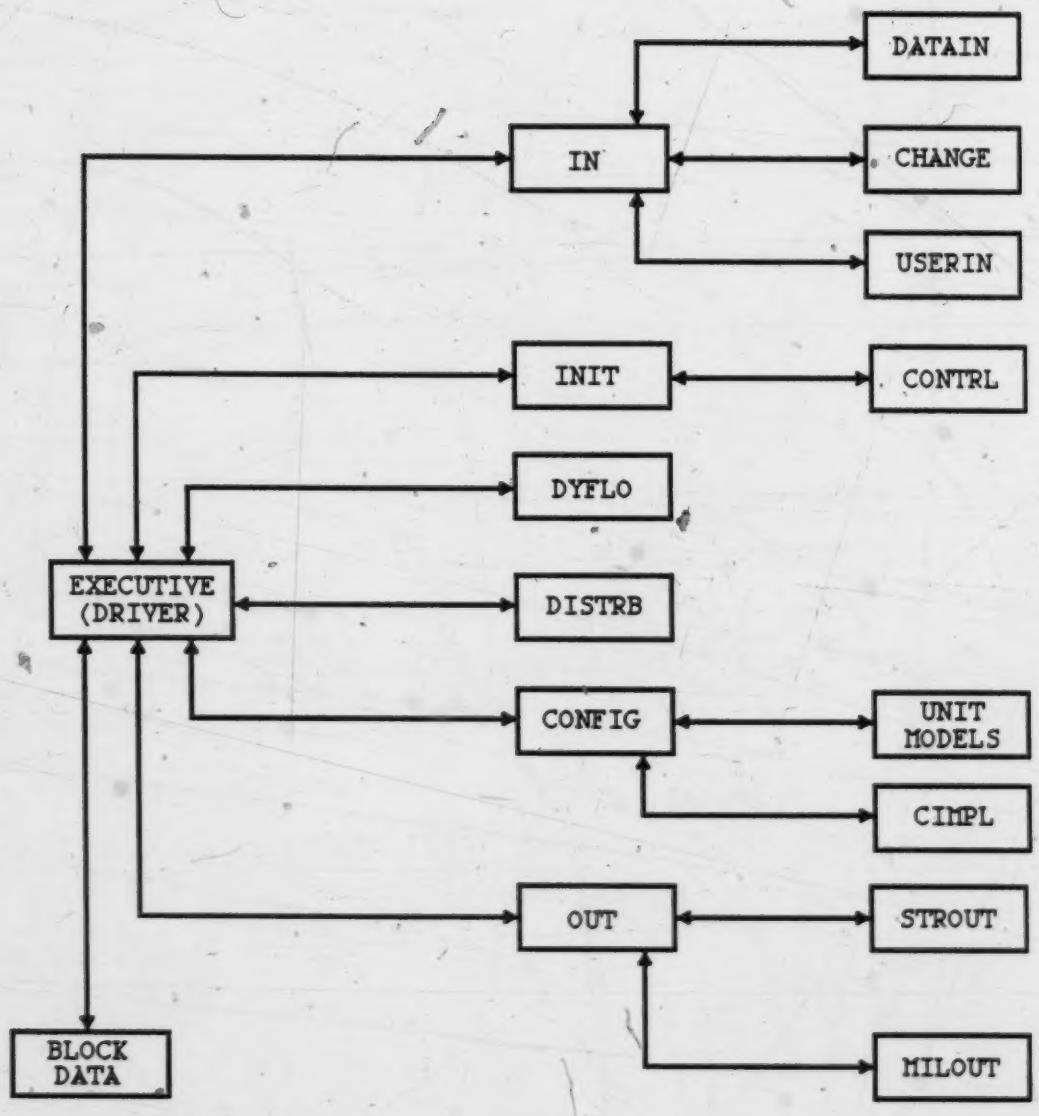


Figure 4.3 Schematic Diagram of the Closed-Loop Simulator Structure

the three parameters that define a first-order system, the time constant τ, the process dead time t_d and the static gain K. Thus the process transfer function $G_p(s)$ can be described in the Laplace domain as:

$$G_p(s) = \frac{Ke^{-t_d s}}{(\tau s + 1)} \qquad (4.3)$$

Smith and Corripio (1985) outline three different techniques for determining these values. Once the transfer function parameters are estimated, various empirical tuning formulae can be used to determine the controller settings, see Smith and Corripio (1985), or Stephanopoulos (1984). It has been this author's experience that if the process is truly first order these methods do in fact give good initial estimates of the controller gains and with some fine tuning on-line provide adequate control of the process. For the grinding circuit simulator, the process reaction curves as indicated in the previous chapter give the appearance, not of a first order with time delay process, but rather as an overdamped second (or higher) order process with time delay. To date there doesn't appear to be a general technique for deriving controller constants for higher order processes so an ad hoc technique has to be used. For the purposes of this work initial tuning was performed using proportional control only, with the controller gain being adjusted until offset in the controlled variable and oscillations in the input and output signal were minimized. The resultant gain was then reduced

by 10-20% and the amount of integral action slowly increased to reduce the offset to zero without causing stability problems. Further fine tuning was performed using the Simplex Method as described by Box et al. (1969) using the chosen performance criteria as the response. The power of the simulator is only fully realized when this type of work has to be done. To tune one loop requires up to a dozen runs with the simulator, with each run taking a few minutes on the University of Alberta mainframe computer, whereas to perform similar tests in the plant would take weeks or months, with the additional cost penalty of plant test far outweighing those of the simulator. Plant tuning problems are additionally compounded by process and measurement noise, a factor that is avoided in the simulations. This technique then has realized various combinations of PI controllers for different situations that provide, if not optimal, then at least adequate control of the circuit for given disturbances.

In order to evaluate the controller performance, some performance criteria has to be chosen. The simplest are based on the characteristics of the closed-loop response of the system and include overshoot, rise time, settling time, decay ratio, etc., see for example Stephanopoulos. The weakness of these criteria is that they don't account for the full dynamic response of the system and minimizing any one of them tends to increase one or more of the others. As a result time-integral performance criteria have been

developed such that the complete response can be used in evaluating the controller performance, see Stephanopoulos page 302. For the purpose of this work, the integral of the absolute error (IAE) is used as the performance index, where

$$IAE = \int_0^\infty |e(t)|\, dt \qquad (4.4)$$

Although the choice of the performance criterion has an affect on the controller design, the use of IAE in this particular application is felt to be quite appropriate.

4.3 Implementing Classical Control on a 2x2 System

From Figure 4.2 it is apparent that the overall circuit can be described as a 3x3 MIMO system. However, from the point of view of developing a control scheme, it is more expedient to examine the simpler 2x2 system first. Once this system has been fully analyzed and the controller interactions recognized and dealt with, expansion to the 3x3 system should be achieved with very little extra effort. For the purposes of this work, the three outputs to be controlled are the cyclone overflow size distribution (COS), the sump level (SL) and the circulating load (CL), with the three manipulated variables being the fresh ore feed rate (FOF), sump water flow rate (SWF) and pump speed (PS). In order to examine the 2x2 system, only COS and SL will be considered as controlled variables, with CL being monitored in order to identify any potential interaction problems. Initially COS

will be controlled directly by FOF, leaving both SWF and PS as inputs to control SL. Although it is obvious that SWF can be used to control COS, the choice of linking COS to FOF has some valid arguments. This progressive attack on the problem follows roughly the work of McDougall (1987), who studied grinding circuit control problems using a precursor of the current simulator that utilized only the fixed speed pump. A second argument for linking COS to FOF is that this simply provides a logical starting point for investigating the control of grinding circuits. Herbst and Rajamani (1979) indicate that in industrial applications there is not as yet any real preference for Type I or Type II control, which would seem to indicate that there is no general consensus as to which control scheme is better.

The previous work of McDougall was limited by the fixed speed pump model, which in essence removes one input from the system, and in order to avoid underdefining the system, removes one output from the system. As a result only COS and SL were considered as controlled variables, with FOF and SWF as the respective inputs. Given that a variable speed pump model has been incorporated into the simulator it is possible to use it to control sump level. This allows for the control of one more output, in this case circulating load. As a result, two scenarios will be discussed in this chapter, with fresh ore feed rate controlling cyclone overflow size, either sump water flow rate or pump speed can be used to control sump level.

One could argue that sump level control is an unnecessary extravagance as long as the situation is avoided in which the sump is running dry or overflowing. This is certainly true in some situations; either the sump is large enough to absorb significant changes in the fresh ore feed rate, or the hardness disturbances are not severe enough to cause large swings in the fresh ore feed rate. However, for the relatively small grinding circuit being simulated, (64 tonnnes/hr), Figure 4.4 indicates the problem of using a fixed speed pump while FOF is being adjusted to compensate for the effects of a hardness change on COS. With well tuned control it is apparent that the FOF is decreasing in response to an increase in the coarseness of the product stream. After about 30 minutes the COS is responding and the rate of decrease in FOF is slowing down, and one would expect it to eventually settle to a new steady state. However, at about 48 minutes the sump runs dry, at which point the validity of the simulator is in doubt since the pump output is constrained to equal the volumetric flow rate into the sump in order to avoid negative holdups in the sump. From a qualitative point of view it could be argued that in the real system the pump would start surging in this situation, which in itself would degrade the cyclone performance and adversely effect the product size. Whatever the outcome, it is obvious that for the circuit under consideration some form of control of the sump level has to be included so that FOF can be manipulated to control COS. The importance of

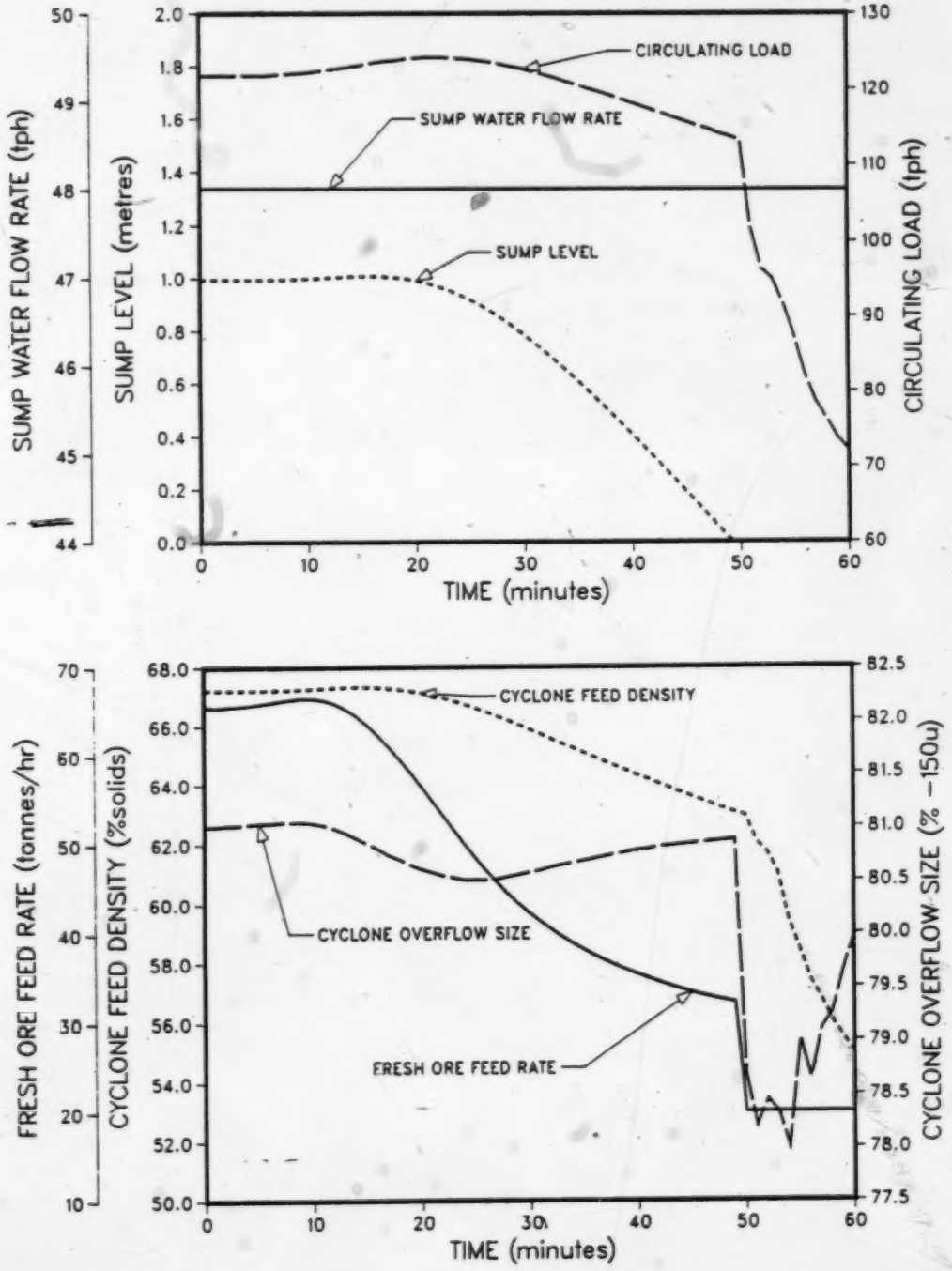


Figure 4.4 FOF Controlling COS, Ore Hardness Increase at t=2 Minutes, Fixed Speed Pump, (K_c=11.95, τ_I=4.07)

sump level can also be considered from another perspective. The sump level can be used as an indication of the inertia of the system; with high sump levels the inertia is increased such that high frequency disturbances can often be absorbed, but on the other hand the system becomes more difficult to manipulate in terms of regulatory control. The opposite is true for low sump levels, the system becomes more susceptible to various disturbances, but easier to manipulate from a control stand point. Once again a balance can be achieved by holding the sump level between these extremes. In addition, by maintaining a fairly constant sump level, the dominant time constants of the units involved in the system will not fluctuate as much as would be apparent with an oscillating sump level. From a control point of view it is much easier to tune a system that has relatively stable time constants.

4.3.1 Controlling Sump Level with Sump Water Flow Rate

The effects of a hardness change on COS is indicated in Figure 3.16. By using FOF to control COS it is obvious that as the product stream becomes coarser FOF should be reduced, and conversely, as COS becomes finer FOF should be increased. Thus as the control action takes effect in response to a hardness change, FOF is reduced, which in turn reduces the volumetric flow of slurry to the sump. Using SWF to control sump level, given that a fixed speed pump is being used, dictates that SWF has to be increased in order to make

up the shortfall in flow to the sump. From Figure 4.5 it is quite apparent that changing the flow rate of water to the sump affects all system outputs. For a 10% decrease in SWF the % -150μ material in the product stream decreases, the cyclone feed density increases, the circulating load decreases slightly and the sump level drops steadily, with the response being reversed for an increase in SWF. Thus it can be expected that if sump water is used to control level, while FOF is being reduced to compensate for the coarsening of the product stream, various interactions occur that could compound the control problem. Since the flow rate out of the sump is basically independent of the sump level, i.e. the effect of the static head is minimal, the response of the sump level to changes in SWF is definitely not first order, and by all appearances has little or no time delay. Given this sort of situation, tuning the sump level controller simply involves choosing a large K_c, and small τ_I to effect fast control without causing stability problems. In order to test the sump level controller a hardness change is introduced to the system with the COS-FOF controller disconnected and the SL-SWF controller active, Figure 4.6. Using K_c=1 and τ_I=0.25 it is apparent that sump level control is excellent. It is useful at this point to compare Figure 4.6 with Figure 3.16 in which a variable speed pump with proportional control was used to control sump level in the presence of a hardness change. Ignoring the offset in sump level due to the proportional control, it is more important to compare

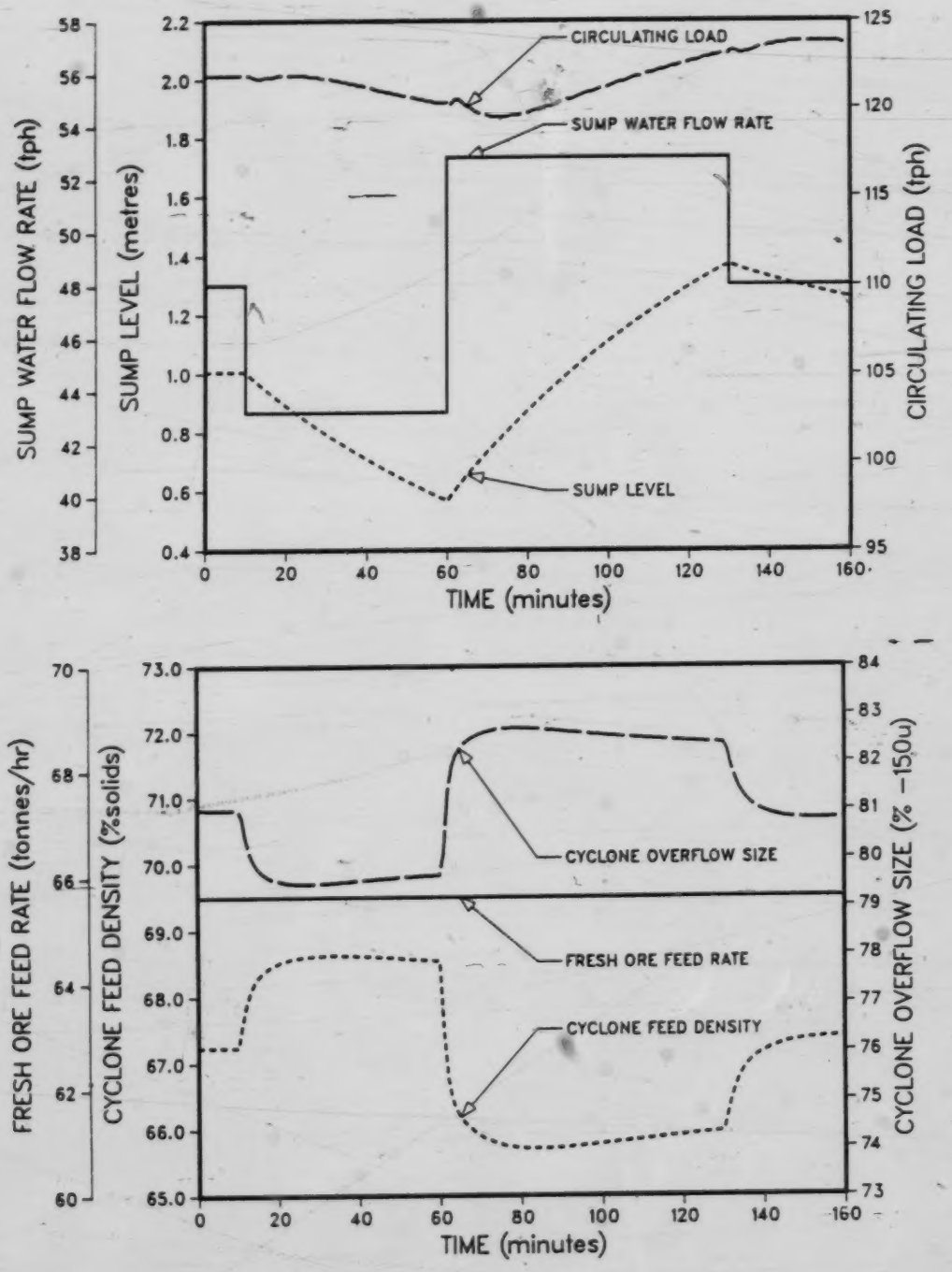


Figure 4.5 Open-Loop Response of Grinding Circuit to Changes in Sump Water Flow Rate

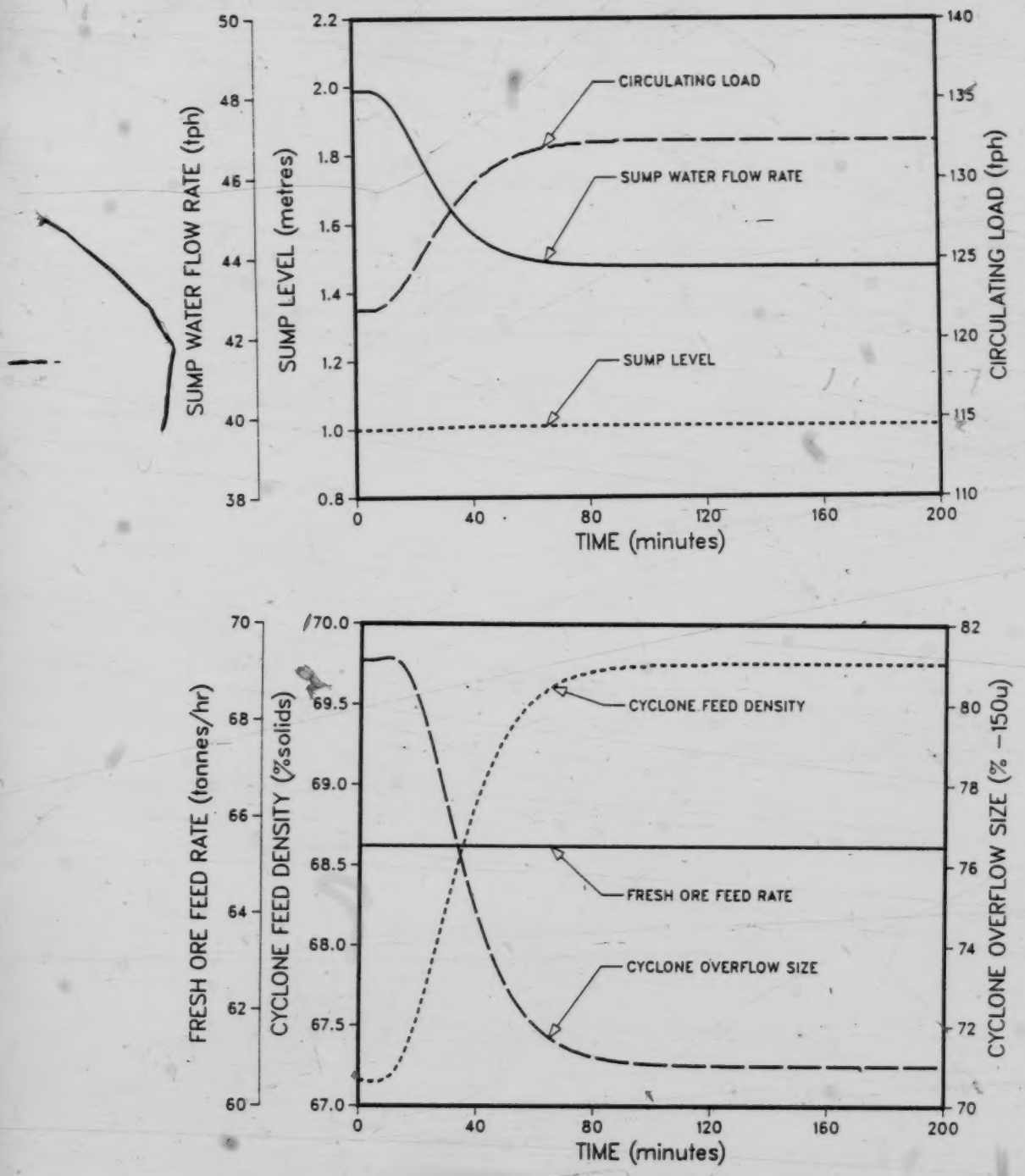


Figure 4.6 Circuit Response to an Increase in Hardness at T=2 Minutes with Sump Water Flow Rate Controlling Sump Level (K_c=1, τ_I=0.25)

the cyclone overflow size and circulating load curves. Using SWF to control sump level results in a slightly lower steady state COS, (or slightly higher cyclone feed density), and a significantly lower steady state circulating load. In addition there is almost a 10% reduction of SWF in order to control the sump level.

It is immediately apparent then that the method used to control sump level has a profound affect on the performance of the circuit. These responses can all be related to the hydrocyclone model used in the simulator in which d_{50c} is a function of the volumetric flow rate to the cyclone, the cyclone feed density and the mass flow rate of water to the cyclone. By altering pump speed to control sump level the volumetric flow rate to the cyclone is the variable that undergoes the most adjustment, whereas the cyclone feed density is more susceptible to change when SWF is used to control sump level. In both cases, the mass flow rate of water to the cyclone also undergoes changes which in turn affects the classifying action of the hydrocyclone. As a result it can be postulated that the root of the interaction problems lie in the classifier model, and it could be that the strength or weakness of the entire simulator is related to the validity of the particular model chosen for the hydrocyclone. Thus for any given action or disturbance in the circuit, one or more of the independent variables that influence d_{50c} may be altered. Depending on the nature of the changes in these variables, they may compound to push d_{50c} in the same

direction, either up or down, they may offset each other exactly such that d_{50c} remains relatively constant, or simply offset each other to varying degrees such that there is some net change in the value of d_{50c}. Whatever the actual response, it is clear that the relationships are quite complex, and when the responses are being analyzed it should be kept in mind that these interactions do exist.

From Figure 4.4 it is apparent that good COS control is achieved, until the sump runs dry, with K_c=11.95 and τ_I=4.07, and from Figure 4.6, using K_c=1 and τ_I=0.25, good control of the sump level is achieved. If the system were non-interacting one could expect to implement both controllers simultaneously and achieve good control of both COS and SL. Suffice to say this is not the case. Running the simulator with the same controller constants as above results in increasing oscillatory behaviour of the inputs and outputs. The integral of the absolute error (IAE) for this run was about 11900 compared to values of 14.6 and 20 for the two tuned closed-loop simulations that are presented here. The interactions are so great that within 30 minutes the inputs oscillate between the limits of 20 and 120 tph for FOF and 0 and 100 tph for SWF. Obviously this is a highly undesirable situation that calls for some form of detuning of the controllers to circumvent the interaction problem.

Various combinations of the controller constants were examined in an attempt to implement good control of this system. In addition, both proportional and PI control of

sump level were evaluated to determine if any advantages could be gained by implementing looser control on the sump level. Figures 4.7 and 4.8 represent the best, i.e. minimum IAE, control of COS by manipulation of FOF with both P and PI control of SL using SWF. In the initial case, P control of sump level, there is the anticipated offset in the final steady state sump level. However, if this is of no concern then the control of COS is seen to be very good, IAE=14.6, with the maximum deviation from steady state of less than 0.5%. By including integral control action on the sump level, the offset is removed, but oscillations in both the inputs and outputs are experienced. Both the controller gain on sump level and the integral action on cyclone overflow size had to be reduced slightly to improve the overall system performance. The cost of this detuning is a slight loss of performance with IAE increasing to 20.0, and the maximum deviation from setpoint increasing to 0.7%. There is the additional consideration when integral action on the sump level is included in that there is an increase in the action of the final control elements, variable speed feeders for FOF and valves for SWF. Although it would not appear to be critical at this stage, it is certainly a point to consider when implementing control in the industrial application. Other comments that can be made at this stage involve the comparison of the final steady state values of the circuit inputs and outputs. In both cases the steady state FOF's are similar at just over 44 tph, and the final cyclone feed

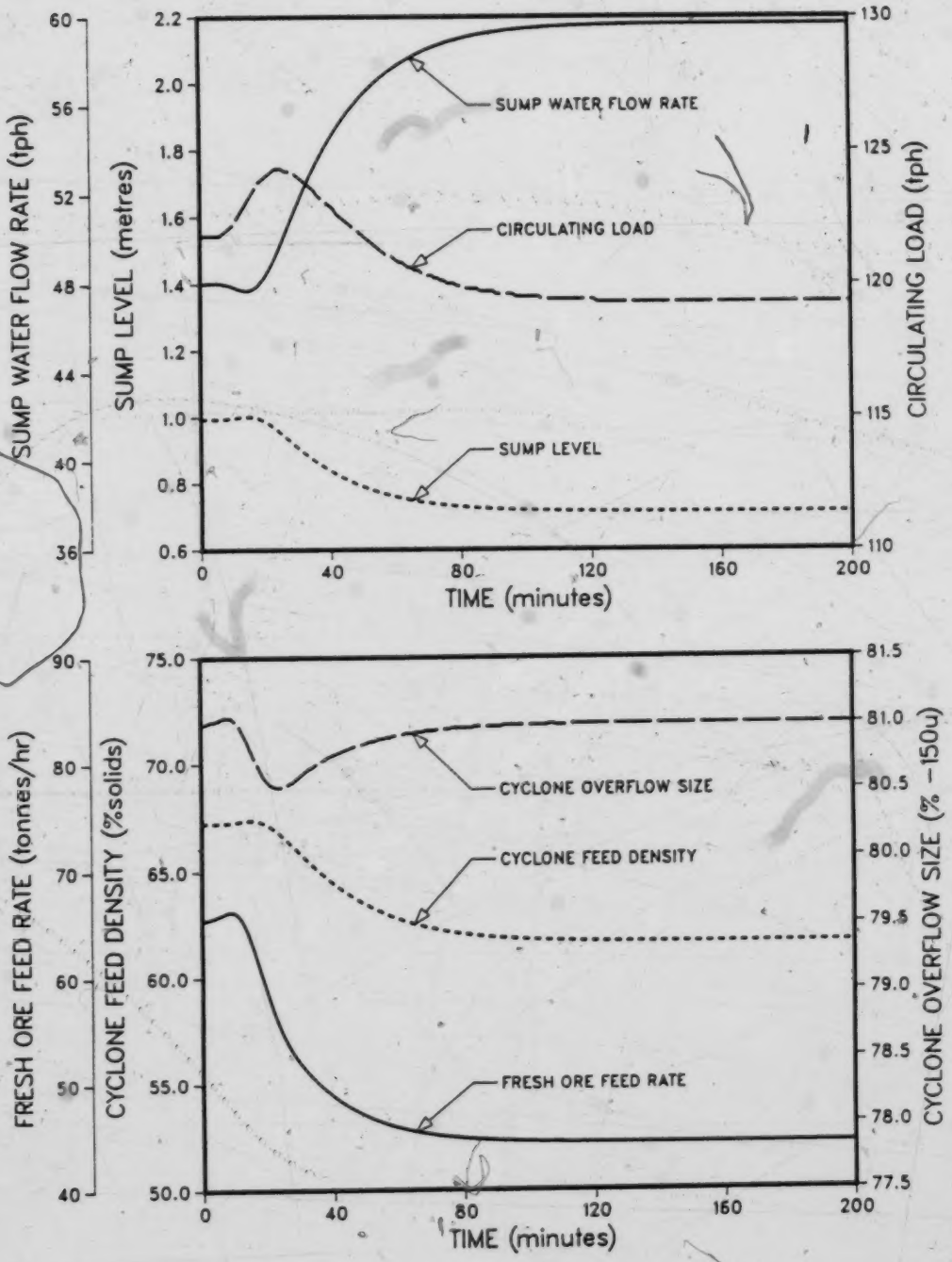


Figure 4.7 FOF controlling COS (K_c=10.5, τ_I=7) and SWF controlling SL (K_c=1), IAE=14.6

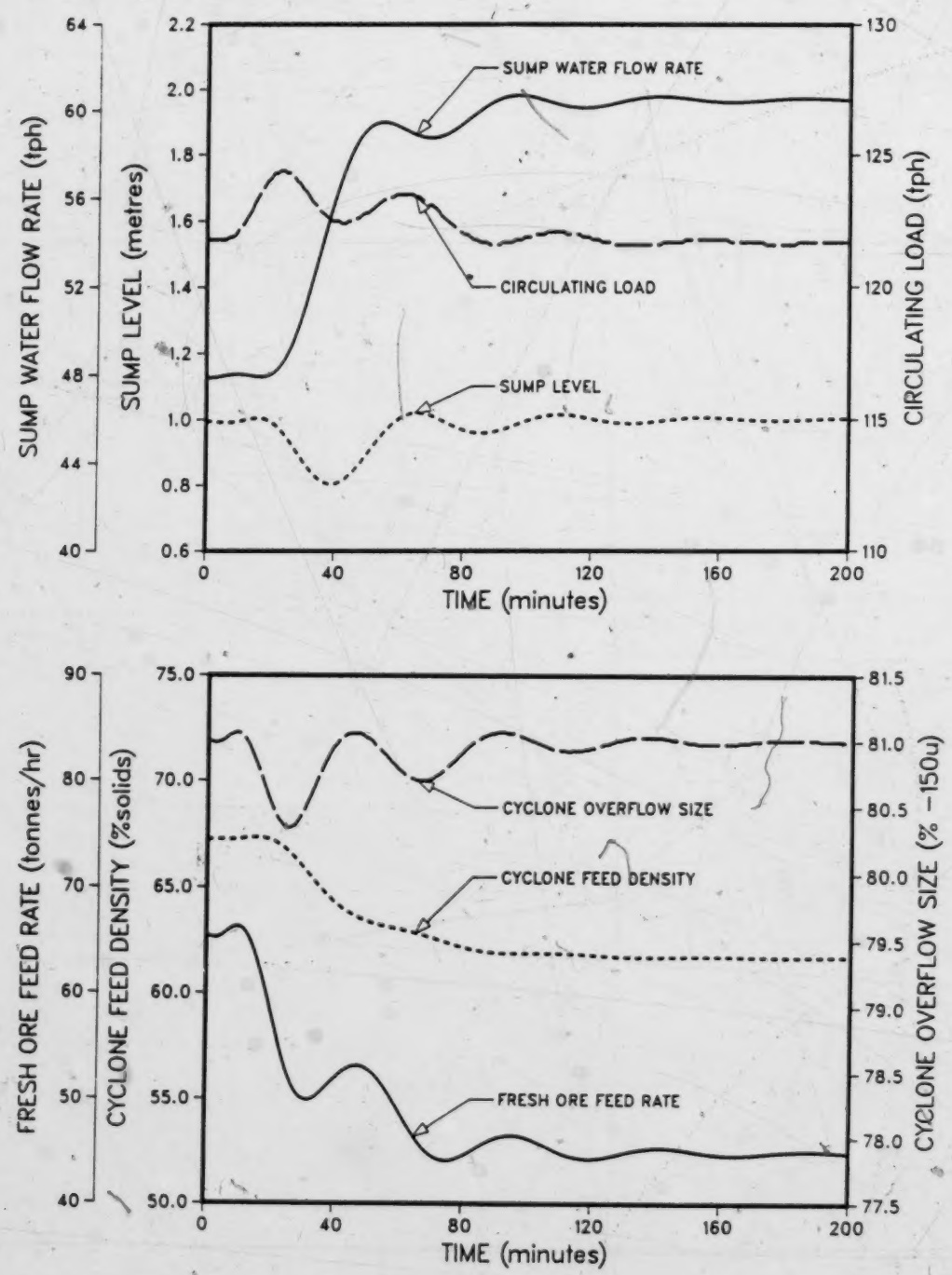


Figure 4.8 FOF controlling COS (K_c=10.5, τ_I=8) and SWF controlling SL (K_c=0.25, τ_I=4), IAE=20.0

density values are also comparable at just over 63% solids. With integral action included the sump water flow rate and circulating load attain slightly higher levels.

4.3.2 Controlling Sump Level with a Variable Speed Pump

The next scenario to be considered is that in which pump speed can be used to control sump level, which will provide a third input to the system so that a third output can be controlled. As in the previous section, it is necessary to get a feel for the possible interactions that will result from manipulation of pump speed, Figure 4.9. Comparing Figure 4.9 to Figure 4.5 it is clear that the cyclone feed density and sump level respond in a similar manner to the changes in the inputs. The major differences are in the response of the circulating load and cyclone overflow size distribution. Of these two, it is the behaviour of COS that will cause the major control problems. As the pump speed is increased there is a sudden jump in COS followed by an exponential decay to a new steady state, with this new steady state being less than the initial value of COS. Such behaviour is called inverse response or nonmimimum phase response, and is the result of two opposing effects within a second or higher order system. Fortunately pump speed will not be used to control COS, but such behaviour will often tend to compound interaction problems.

In order to understand this behaviour it is necessary to quickly review the hydrocyclone model presented in the

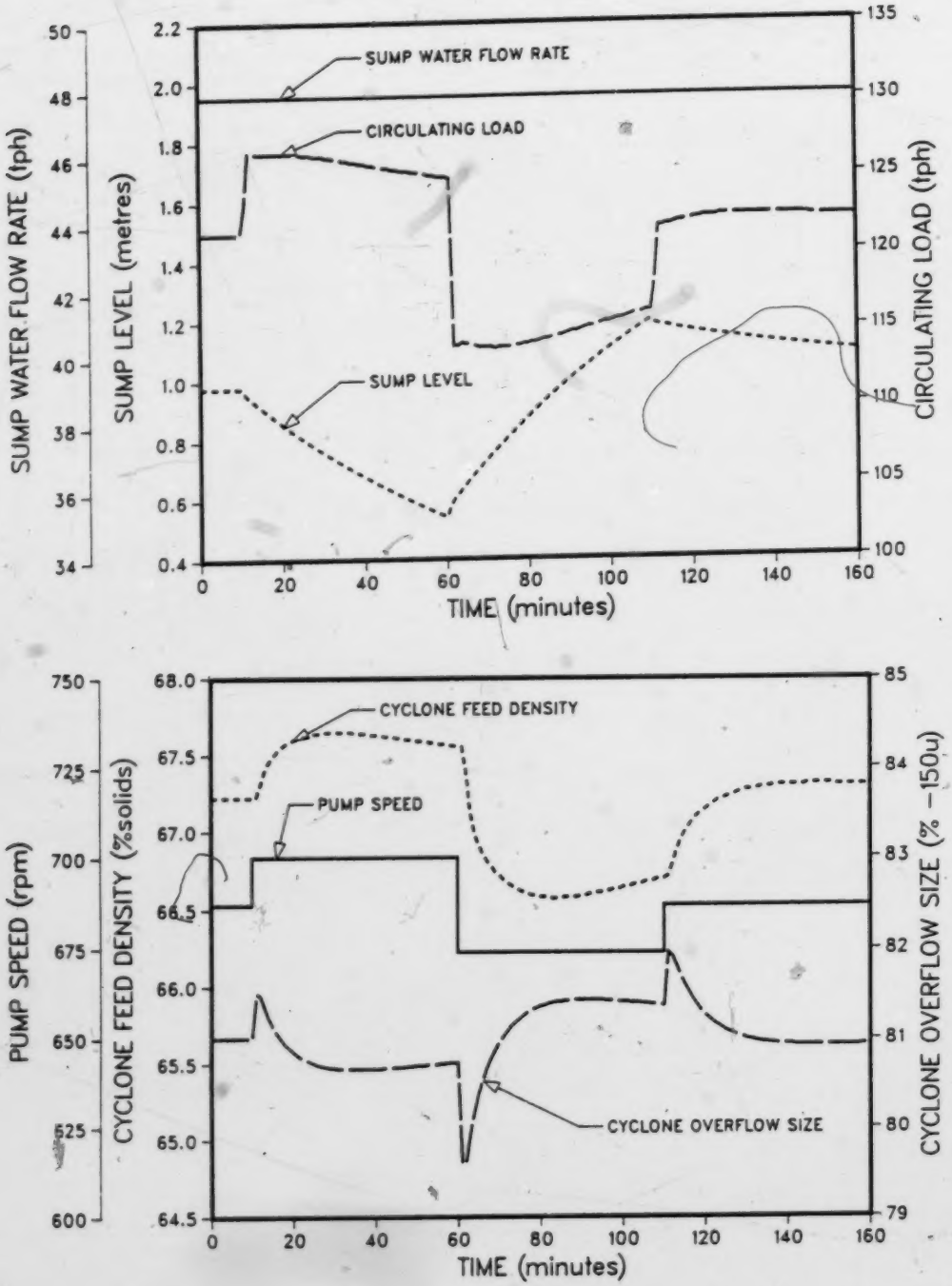


Figure 4.9 Open-Loop Response of Grinding Circuit to Changes in Pump Speed

previous chapter. By taking equation 3.15 and substituting for pressure from equation 3.13 and W_o from equation 3.14 yields

$$\log_{10}(d_{50c}) = 3.0544 + 0.0145\left[\frac{Q}{32.42\ F_{wf}^{1/8}}\right]^2 - 0.015\ W_f \quad (4.5)$$

In this case, in the short term, the sudden increase in volumetric flow rate caused by the step up in pump speed increases d_{50c} while the increase in the mass flow rate of water in the cyclone feed (W_f) forces d_{50c} down, with the net result being a reduction in d_{50c}, routing more material to the underflow (hence the sudden increase in circulating load) which in turn causes the product stream to become finer. As the circulating load builds up, the solids mass flow to the sump increases, and with all other factors being equal, the cyclone feed density increases. The mass fraction of water in the cyclone feed (F_{wf}) decreases, and most probably W_f decreases slightly, which all combine to increase d_{50c} and coarsen the product stream. The end result is that the % -150μ material in the products stream achieves a new steady state below that recorded at the lower pump speed. For a step decrease in pump speed it would appear that the mechanics are simply reversed.

Although diverging slightly, it is of interest at this point to contrast the Lynch-Rao d_{50c} model with that of Plitt (1976) because it would appear that the cyclone model is of paramount importance to the validity of the simulator.

The Plitt d_{50c} model accounts for the cyclone geometry as well as the slurry characteristics. Lumping the variables that are not germane to this discussion into a single constant in both the numerator and denominator, the Plitt d_{50c} equation is of the form

$$d_{50c} = \frac{K_1 \exp(0.063\phi)}{K_2 Q^{0.45}} \qquad (4.6)$$

where K_1 and K_2 are constants, Q is the cyclone feed volumetric flow rate and ϕ is the volumetric concentration of solids in the cyclone feed slurry. If this equation were to be used in the simulator, and a step increase made in the pump speed then Q would increase causing d_{50c} to decrease. Then as the circulating load builds up, ϕ would increase which would tend to force d_{50c} to increase. Although only a qualitative comparison, it would appear that this equation moves d_{50c} in the same manner as the Lynch-Rao model, but for totally different reasons. The major difference is that the Lynch-Rao model involves a Q^2 term in the numerator of a $\log_{10}$ transformation whereas the Plitt model uses $Q^{0.45}$ in the denominator of a linear transformation. Combining these factors makes the Lynch-Rao model more susceptible to changes in the operating variables, and possibly compounds some of the control problems experienced with the simulator. Since not enough information regarding cyclone geometry is available from the work of Smith and Guerin it is necessary to retain the current models, but it would certainly make

for an interesting exercise to compare the performance of the simulator using the two types of cyclone models.

The relationship between sump level and pump speed as indicated in Figure 4.9 provides the basis for designing a controller. Since excessive manipulation of pump speed to produce tight level control is expected to cause further control problems, and since the sump level can safely fluctuate within certain limits, it is only necessary to implement proportional control. From Figure 4.9 it is obvious that the pump speed/sump level relationship is not that of a first order plus time delay system. As such, none of the empirical tuning techniques mentioned so far are of any real use. However, it turns out that this is a very simple loop to tune and Figure 4.10 shows the controller performance given a 20% increase and decrease in the sump water flow rate using a proportional gain of 2.

It is now possible to run further simulations using pump speed to control sump level and fresh ore feed rate to control the product stream size distribution in the presence of an increase in the ore hardness. Some detuning of the COS controller has to be made in order to effect suitable control, but even the best performance with tight level control is substantially worse than that experienced in the previous section. From Figure 4.11 it is apparent that the maximum deviation from the COS setpoint is 2.5% and the IAE is 113.3, compared to 0.5% and 14.6 respectively as displayed in Figure 4.7. In tuning this particular configuration it

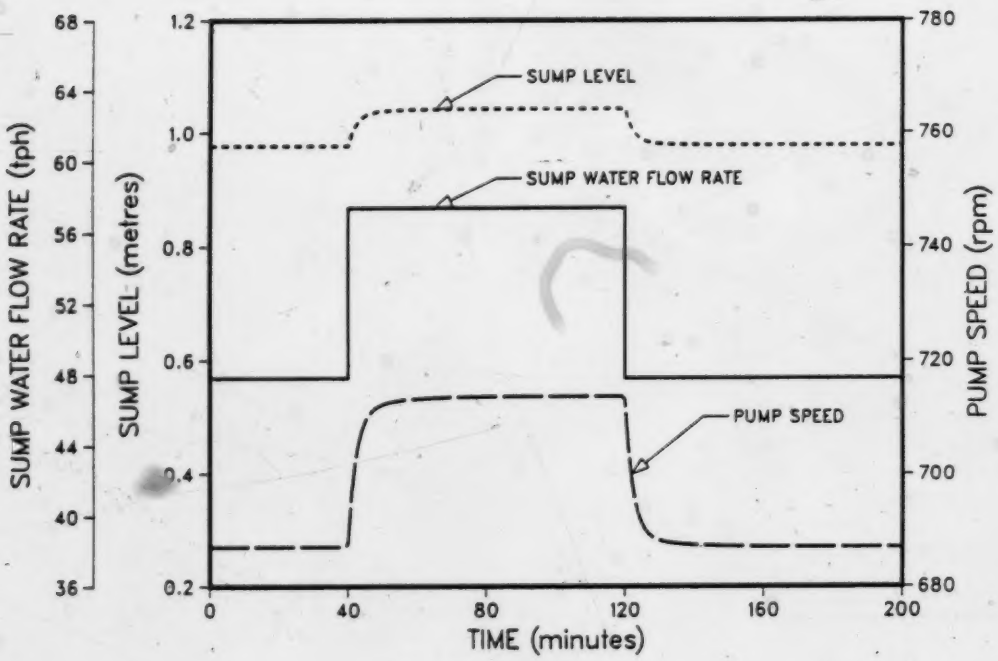


Figure 4.10 Control of Sump Level by Manipulation of Pump Speed in Response to a Step Change is Sump Water Flow Rate

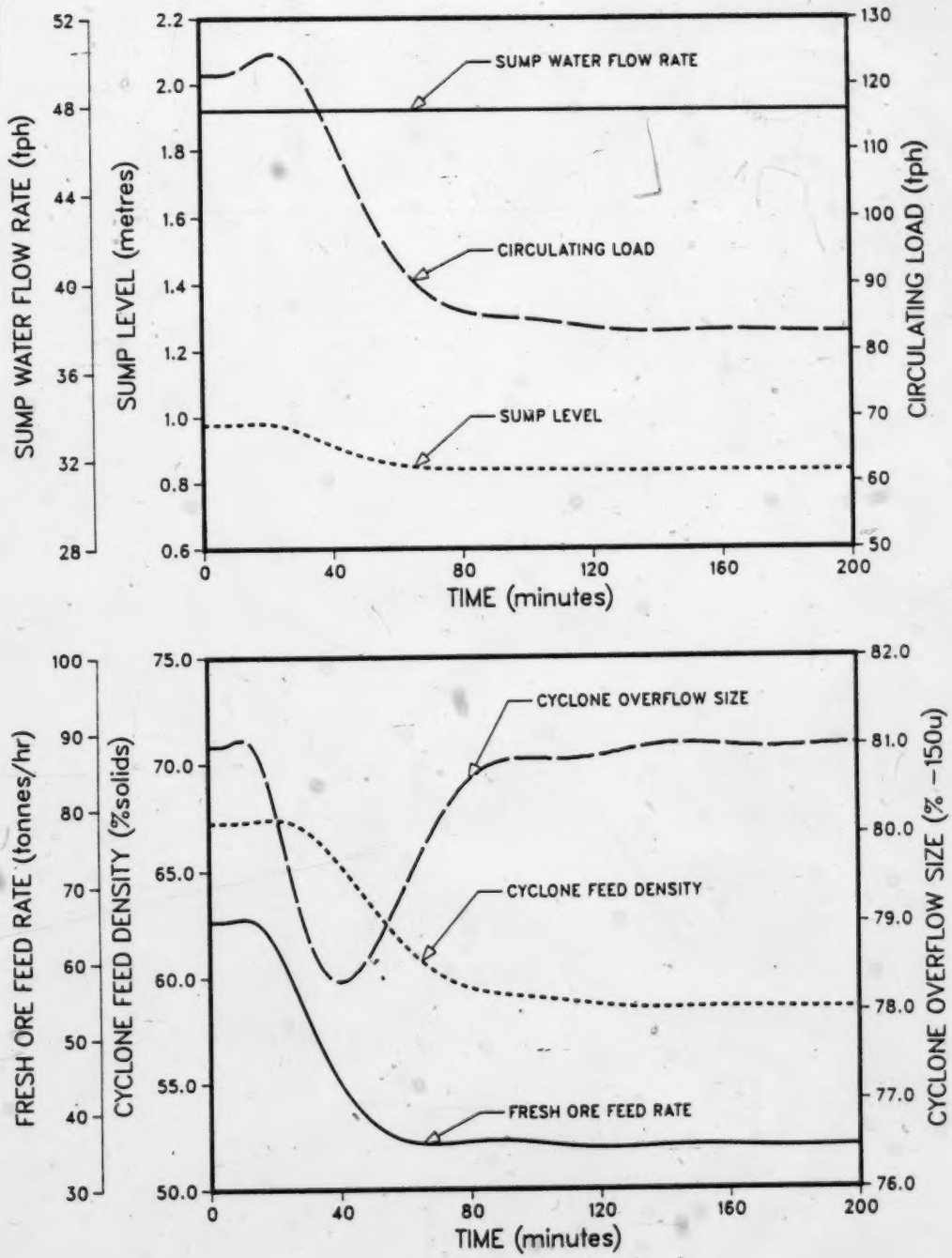


Figure 4.11 FOF controlling COS (K_c=4, τ_I=15) and PS controlling SL (K_c=2,), IAE=113.3

was noted how susceptible the system stability was to the magnitude of the controller gains. By using values of K_c=5 and τ_I=10, which in effect doubles the amount of integral action on COS, and maintaining K_c=2 on the sump level, the result is increasing oscillations in both the input and output variables, Figure 4.12. One of the reasons for including this particular simulation is to stress the importance of understanding the performance function IAE. Initially in tuning the controllers, runs of 100 minutes were made. Suprisingly, using K_c=5 and τ_I=10 gave the minimum IAE over 100 minutes. However, not until the simulation was expanded to 200 minutes were the full effects of these controller gains felt. So, as with most objective performance indices, some subjective judgement has to be used to fully determine the validity of the results.

Since the sump level offset with tight proportional control is less than 20%, it is quite feasible to back off on the level control in order to tighten up on the COS control and improve the overall circuit performance. By decreasing the sump level controller gain to 0.5, the COS performance is greatly improved. Using K_c=14 and τ_I=7, i.e. tripling the proportional action, and increasing the integral action eightfold, reduces the IAE to 19.5 which is comparable to the control achieved using sump water flow rate to control sump level, see Figure 4.13. The penalty for this improved COS performance is that the sump level offset is now 50% which may be close to the upper limit of acceptance

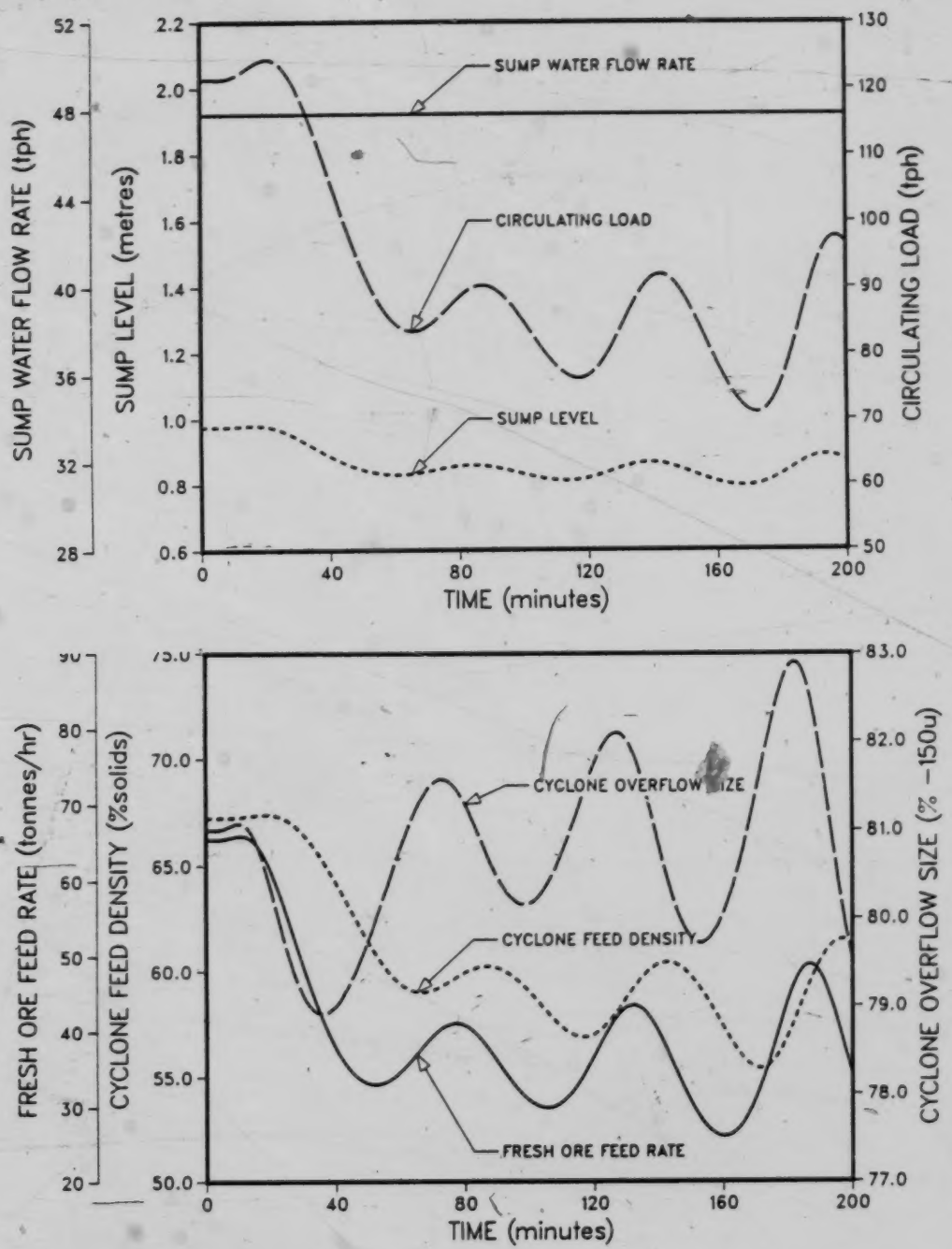


Figure 4.12 FOF controlling COS (K_c=5, τ_I=10) and PS controlling SL (K_c=2,), IAE=161.3

for this particular output. Another point to be made at this time is that the final steady state FOF value is quite a bit lower than that experienced using SWF to control level, (37 tph ys 45 tph). This is an important consideration if the plant throughput has to match the mine production rate, and will undergo further scrutiny when the Type I and Type II control schemes are examined in the next chapter. With the sloppy level control the COS performance becomes less sensitive to the magnitudes of the controller gains, within limits of course, indicating the increased stability of the system.

To further investigate the effect of pump behaviour on COS control, a simulation was run with perfect level control, i.e. the volumetric flow rate out of the sump at each iteration was simply set equal to the volumetric flow rate into the sump. Interestingly enough the COS performance was not further degraded from that experienced with the tight level control. This would seem to indicate that the relationship between the sump level gain and the COS performance is highly non-linear. An interesting exercise would be to plot the minimum IAE achievable versus the sump level controller gain in order to determine the maximum value of the sump level K_c at which acceptable COS performance is realized. Although computationally expensive, but cheap compared to plant tests, this type of exercise could shed further light onto the complexities of the circuit.

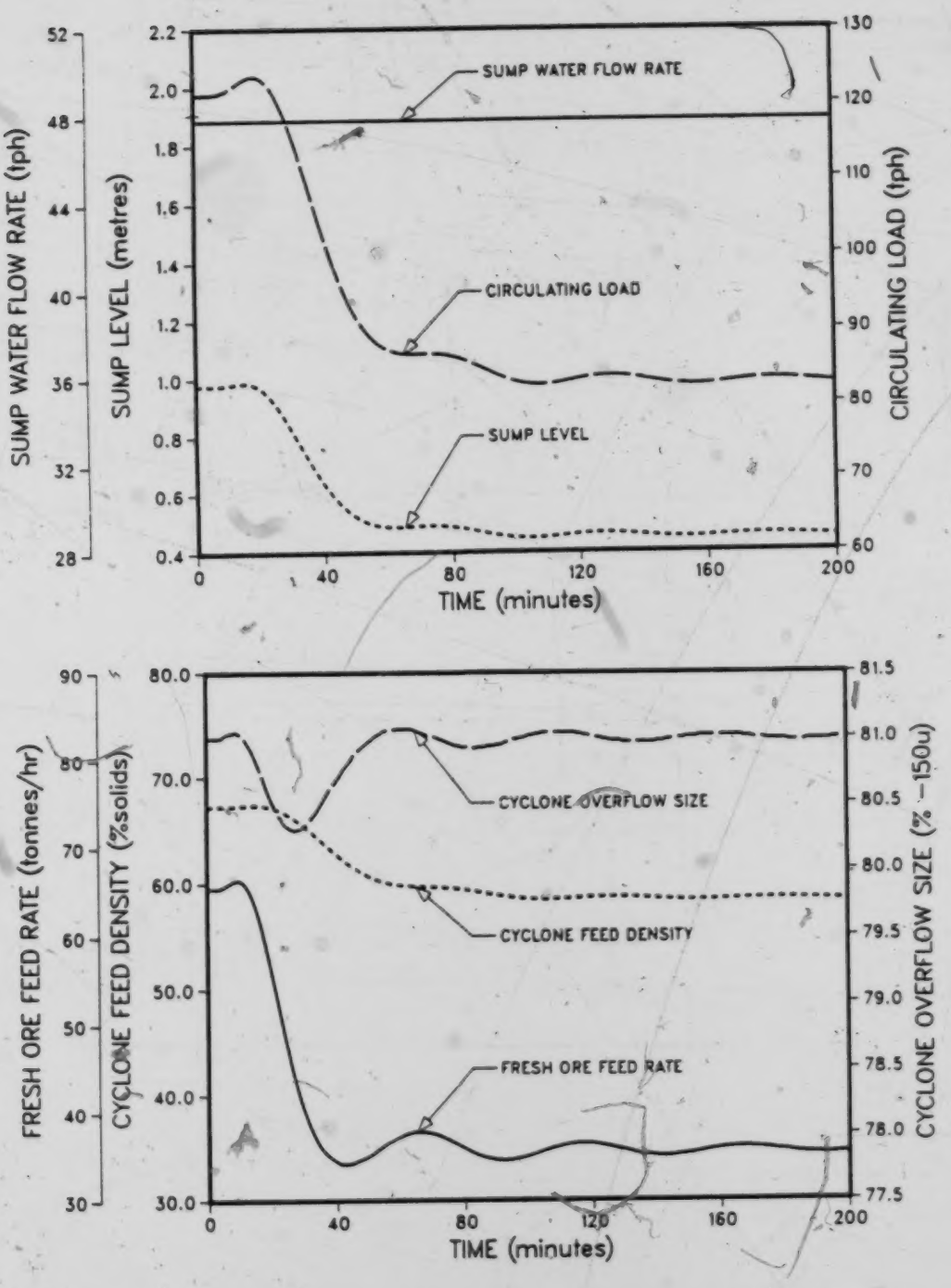


Figure 4.13 FOF controlling COS (K_c=14, τ_I=7) and PS controlling SL (K_c=0.5,), IAE=19.5

Although the controller interaction problems experienced in tuning this particular loop were quite frustrating, it is gratifying to note that other workers have experienced similar problems. Lynch (1977) indicates that PI level control using a variable speed pump can have significant destabilizing action on the short term performance of a grinding circuit. In fact the essence of Lynch's discussion seems to verify the procedure used by this author in implementing good COS control. His suggestion is to use only proportional control, and set the gain as small as possible consistent with avoiding extremes in level under transient conditions, which is basically what has been done in this work. Often the sump level signal can be fairly noisy due to the turbulence in the sump, so another suggestion is to filter these high-frequency components so that the pump speed doesn't react to these transients.

4.4 Summary

On the basis of the work performed in this chapter it is apparent that good control of COS can be implemented using either sump water flow rate or a variable speed pump to control sump level. In both cases, improvements could be realized in the COS controller by detuning the sump level controller. Although there are some disadvantages in using pump speed to control sump level, i.e. greater interaction problems, lower FOF at steady state and some reduction in the performance of the COS controller, these are felt to be

outweighed by the major benefit which is the addition of another input. This in turn allows for the control of an extra output, in this case circulating load. The Type I and Type II control systems to be studied in the next chapter will maintain good COS control in the face of a hardness disturbance, but at the same time hold the FOF higher so that plant capacity is not lost. In all future work the variable speed pump will be used to control sump level using a proportional gain of 0.5, and as such may not be be indicated explicitly for all the simulations.

The true power of the simulator is not fully appreciated until the problem of controller tuning is addressed. In order to tune the loops discussed in this chapter it is estimated that over one hundred simulations had to be made. To perform similar work at the plant level would probably be deemed unfeasible. Although the computational costs on the University of Alberta mainframe are quite high, using just over 20 seconds of CPU time for a 200 minute simulation, and the simulator is quite a bit slower when run on the PC equipped with a math coprocessor, (about 20 minutes for a 200 minute simulation, perhaps ten times slower with no coprocessor installed), these costs are minimal when compared to the costs of even a single plant level experiment. With the batch run utilities now available on most operating systems it is possible to set up multiple runs on the PC and allow the machine to run overnight, so the inconvenience is really not that great.

However, and this point cannot be overemphasized, it is very important that the simulator results be treated as no more than good estimates of what will occur in the industrial situation. The simulator shortens the lead in time to any plant investigation, and is extremely useful in highlighting unusual characteristics of the circuit that may be overlooked otherwise. Although it could be argued that some of the models may require further verification with plant data, from an educational point of view the simulator is felt to accurately portray the response of a grinding circuit to various disturbances and changes in inputs. By analysing the behaviour of the circuit it is possible to garner a greater understanding of the circuit behaviour and to fully appreciate the magnitude of the control problems that face the mineral processing engineer.

5. Type I and Type II Control

From the previous work it is apparent that the fresh ore feed rate to the circuit can be manipulated in order to control the cyclone overflow size distribution in the presence of a change in ore hardness. By using a variable speed pump to control sump level, good control of COS was achieved, but at the cost of a major drop in FOF, from 65.4 tph down to 35 tph, and a significant reduction in circulating load, 120 tph down to 83 tph, see Figure 4.12. The alternate strategy is to use a fixed speed pump and use the sump water flow rate to control sump level. It was shown, Figure 4.7 and 4.8, that this implementation had some advantages in that FOF was only reduced to 45 tph and CL was maintained at much higher levels. However, by using the variable speed pump it was proposed that SWF could be used as an extra input so that Type I and Type II control strategies, (Herbst and Rajamani, 1979), could be implemented. It would then be possible to make comparisons to determine if any further improvements were realized from these control strategies.

This chapter will evaluate the grinding circuit as a 2x2 system with FOF and SWF as inputs and COS and CL as the outputs, with the sump level control by the variable speed pump being considered simply as an integral part of the process. From the open-loop studies in chapter 3, and again with the closed-loop studies in chapter 4, it is evident that manipulation of either of the inputs affects both the outputs. As a result, an analysis of the interactions will

be provided in order to determine the best pairing of inputs and outputs. Based on these results, Type II control is implemented and the performance evaluated in order to quantify the benefits realized. Although the improvements are significant, further improvements are realized through the inclusion of static decouplers. The chapter is completed by showing that the controller performs equally as well with a fixed speed pump which allows the sump level to fluctuate. The full ramifications of this observation are currently unclear as the results appear to contradict the current philosophy of grinding circuit control which indicates that good control cannot be achieved without the use of a variable speed pump to control sump level, Gault et al. (1979).

5.1 Interaction Analysis

Given a system with two inputs and two outputs there are only two possible combinations of control loops, but for an NxN system there are N! combinations. Therefore it would be useful to evaluate the interactions between the inputs and outputs in order to produce control configurations such that the interactions are minimized. The method first proposed by Bristol, (see e.g. Stephanopolous, 1984), known as the relative gain array (RGA) provides such a technique. The RGA method is fairly simple to use and understand and requires only the steady state gain information that is obtainable from the process reaction curve. The major disadvantage is that no consideration is given to the

dynamics of the interactions, however, it has proven to be succesful in numerous industrial processes, (Smith and Corripio, 1985), and is felt to provide a good insight into the expected control behaviour. In essence the relative gain can be defined as the ratio of the open-loop steady state gain to the static gain realized when other outputs are kept constant via some form of feedback control, i.e. the closed-loop steady state gain. These values can be determined analytically if accurate models of the process are known. Since this is not usually the case the values have to be obtained by direct experimentation or through simulation.

For the grinding circuit, the steady state gain matrix, K, can be defined as

$$\begin{bmatrix} COS \\ CL \end{bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} FOF \\ SWF \end{bmatrix} \qquad (5.1)$$

The values of the elements of the K matrix can be obtained from open-loop tests as presented in Figures 5.1 and 5.2. The former indicates the results of a 20% decrease in FOF, to obtain K_{11} and K_{21}, and the latter a 20% decrease in SWF to determine K_{12} and K_{22}. From these figures, or from the tabulated data available from one of the simulator output files, the steady state gains can be calculated, for example,

$$K_{11} = \frac{\Delta COS}{\Delta FOF} = \frac{84.27 - 81.00}{52.50 - 65.63} = -0.249 \qquad (5.2)$$

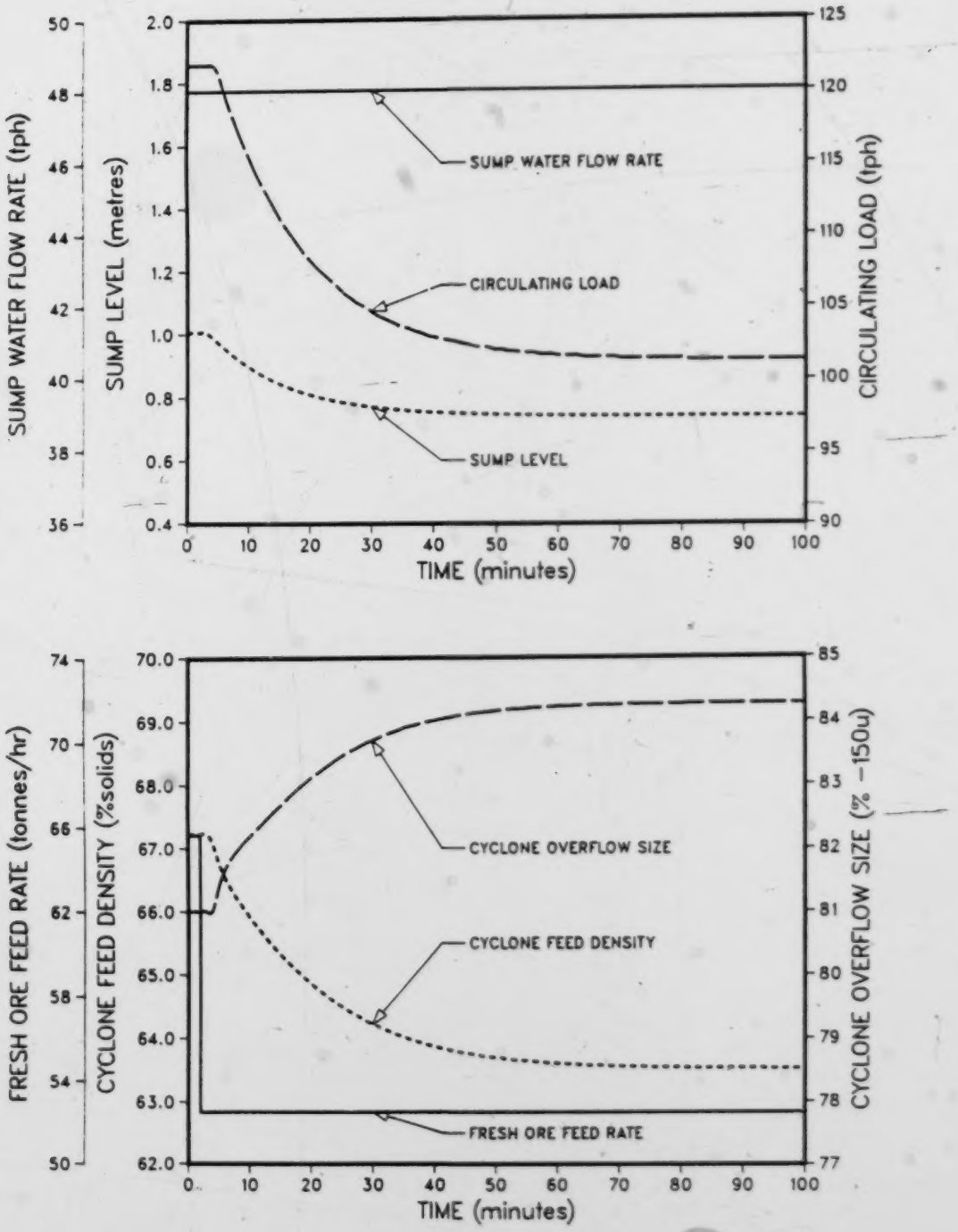


Figure 5.1 Open-loop Response of Circuit to a 20% Decrease in Fresh Ore Feed rate

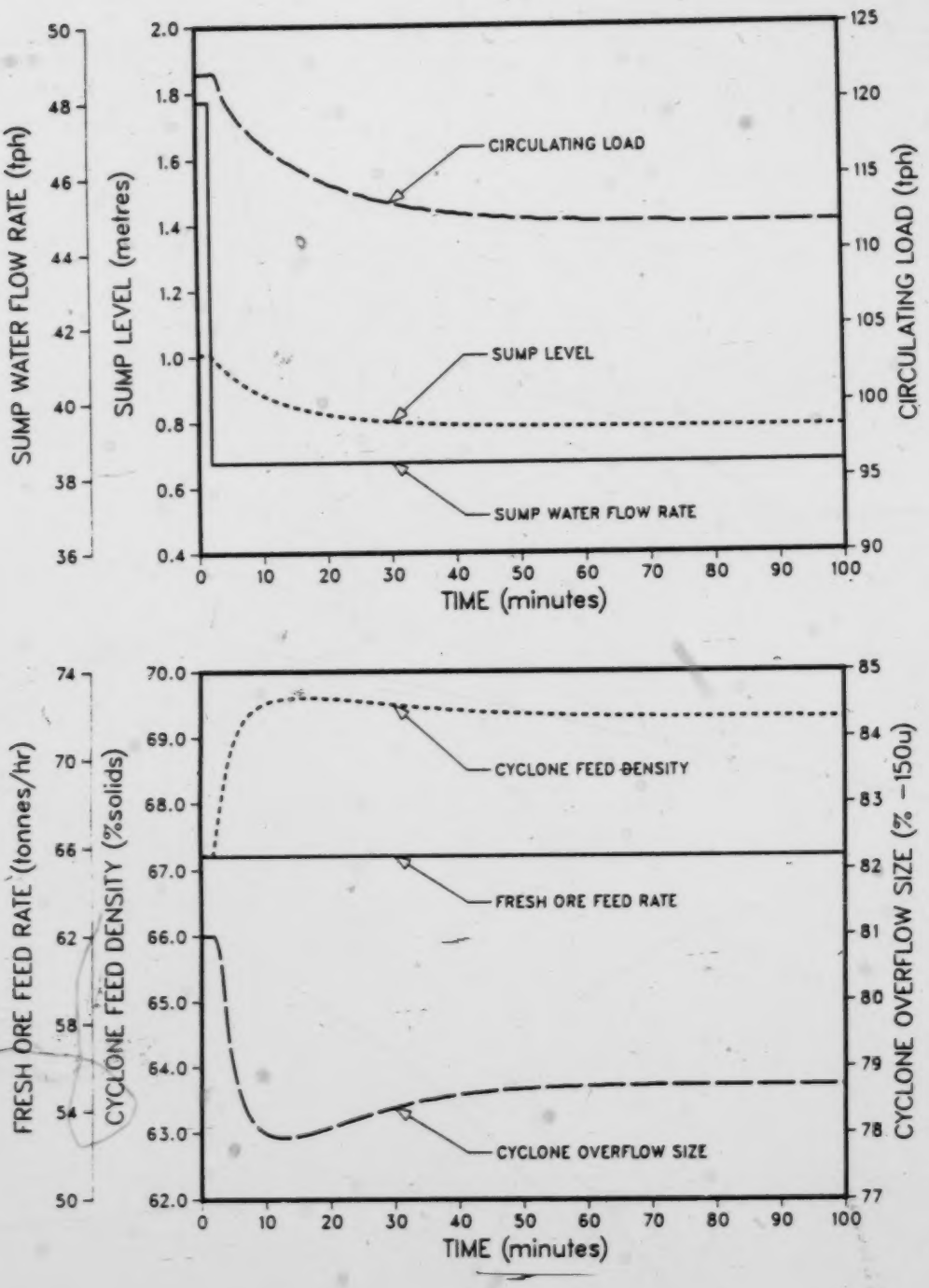


Figure 5.2 Open-loop Response of Circuit to a 20% Decrease in Sump Water Flow rate

similarly,

$$K_{21} = \frac{\Delta CL}{\Delta FOF} = 1.569 \tag{5.3}$$

$$K_{12} = \frac{\Delta COS}{\Delta SWF} = 0.236 \tag{5.4}$$

$$K_{22} = \frac{\Delta CL}{\Delta SWF} = 1.042 \tag{5.5}$$

The closed-loop gains, K'_{ij} can be obtained either through simulations or mathematically. For a 2x2 system the latter method is more expedient since it can be shown, (Smith and Corripio, 1985), that, for example

$$K'_{11} = \frac{K_{11}K_{22} - K_{12}K_{21}}{K_{22}} \tag{5.6}$$

Thus all the information for the 2x2 system can be garnered from the two open-loop simulations presented in Figures 5.1 and 5.2. The elements of the relative gain array λ are simply

$$\lambda_{ij} = \frac{K_{ij}}{K'_{ij}} \tag{5.7}$$

Since the sum of the relative gains in any row or column has to equal unity, for a 2x2 system it is only necessary to calculate one element directly with the others being derived by difference. Thus for the system at hand

$$\lambda_{11} = \frac{K_{11}}{K'_{11}} = \frac{K_{11}K_{22}}{K_{11}K_{22} - K_{12}K_{21}} \tag{5.8}$$

with

$$\lambda_{22} = \lambda_{11} \tag{5.9}$$

$$\lambda_{12} = \lambda_{21} = 1 - \lambda_{11} \tag{5.10}$$

Using the values presented above from the steady state gains, the relative gain array λ is determined to be

$$\lambda = \begin{bmatrix} 0.412 & 0.588 \\ 0.588 & 0.412 \end{bmatrix} \tag{5.11}$$

Both Stephanopolous (1984) and Smith and Corripio (1985) provide a detailed analysis of the meaning of the numerical values of the λ_{ij}. Suffice to say at this juncture that the pairings should be chosen in such a way that the relative gains, λ_{ij}, are positive and as close to unity as possible. From Equation 5.11 it is apparent that the preferred pairing is SWF controlling COS and FOF controlling CL, i.e. Type II control. However, the indication is that there is still a significant amount of interaction with this pairing, how detrimental this will be to the controller performance can only be determined by performing the appropriate simulations.

5.2 Implementing Type II Control

As in the previous chapter, deriving the control constants through the process reaction curve technique resulted in quite unacceptable control performance. Since four parameters are now involved in the tuning process, it is quite a complex procedure to produce effective control performance. The methodology adopted by the author involves bringing one loop under tight PI control, in this case the SWF-COS loop, with only a slight amount of P only control on the second loop, the FOF-CL loop. Once the first loop is tuned, control on it is backed off 10-20% and tightened up on the second loop until the interaction problems start to significantly degrade the performance of the first loop. The final fine tuning stage is very subjective, with changes being made to individual controller gains, both P and I, based on knowledge of the process itself and an understanding of how the gains effect the circuit performance. The integral of the absolute error was monitored for both output signals, but it was deemed that the optimal controller was not the one that minimized the sum of the two IAEs. This is because the COS IAE is measuring deviations in percentages while the CL IAE is concerned with tonnes per hour. It is felt that a deviation of 3% in the COS is much more critical than a deviation of 3 tph in CL. Therefore some weighting must be given to each IAE in order to design the "best" controller. For the purpose of this exercise, the weighting has been fairly subjective since there is no information currently available

regarding potential operating constraints. As such, the performance depicted in Figure 5.3 is just one of the various combinations of controller gains that produced good overall control. With SWF controlling COS (K_c=9, τ_I=5) and FOF controlling CL (K_c=4, τ_I=40), the IAEs were 7.4 and 203 respectively. Although the second IAE appears to be relatively large, examination of the figure indicates that the maximum deviation from set-point is only 3 tph, which by industrial standards is probably quite insignificant, and depending on the method of monitoring CL quite within the range of the instrument accuracy.

A quick comment here is that measuring circulating load in the industrial grinding circuit is a difficult task. In the simulator it is simply a matter of comparing the mass flow rate of stream 14, the ball mill discharge, to the set-point and taking the necessary control action. In industry, direct measurement of the ball mill discharge mass flow rate is usually not possible. One technique is to measure mill load using load cells and/or torque, sound or watt meters and then relate that signal to the circulating load. Unfortunately in overflow discharge ball mills the holdup is relatively constant, i.e. independent of slurry mass flow rate, and as such these signals are limited in the amount of information they can provide. A more sophisticated method is to monitor the cyclone feed stream for both slurry density and flow rate. These two signals can be used to calculate the solids mass flow rate to the cyclone. Since there are

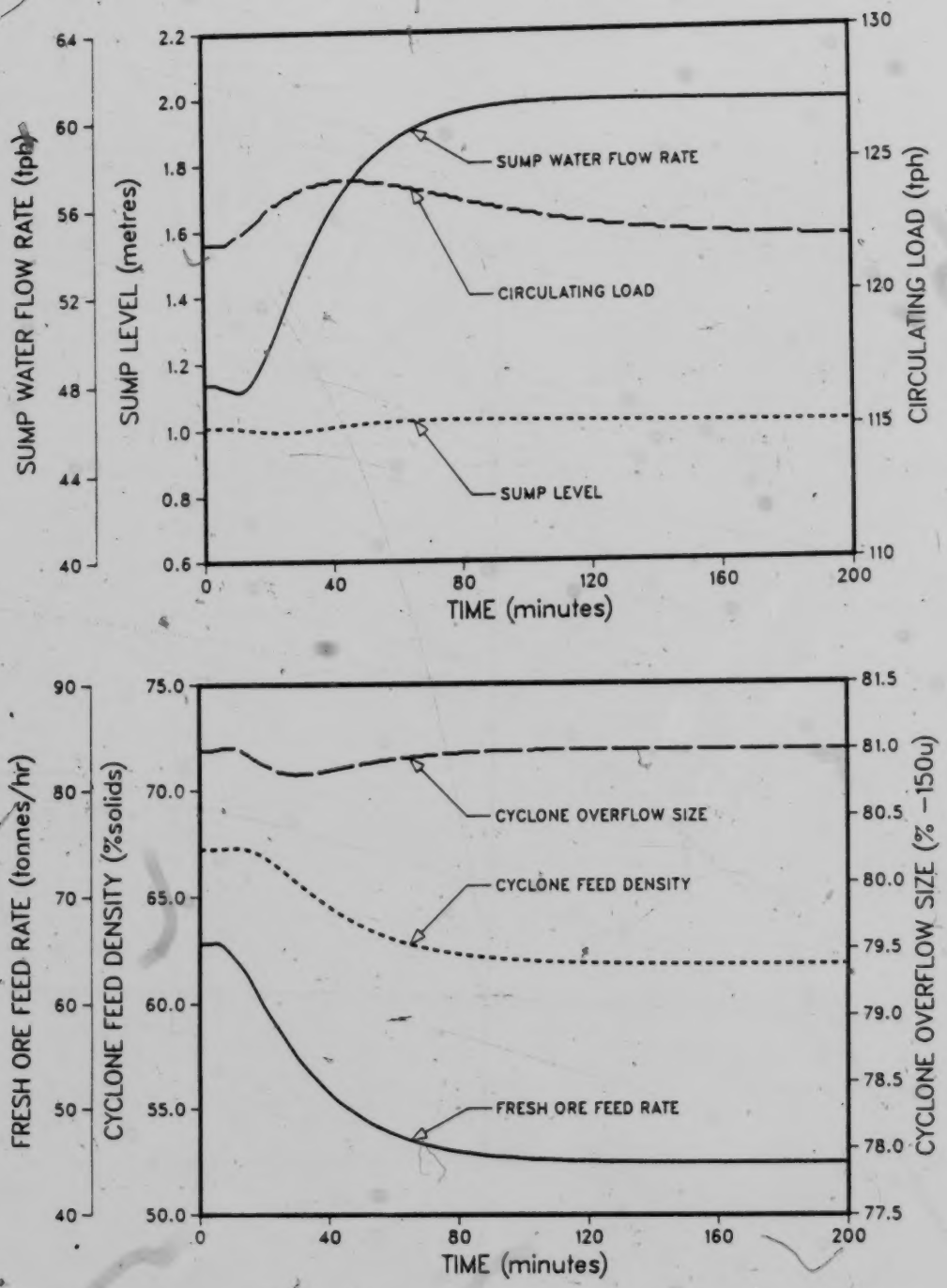


Figure 5.3 Type II Control, Response to a Step Increase in Ore Hardness at t=2 Minutes

only two sources of solids to the sump, the difference between this value and the FOF value from the rod mill feed weightometer indicates the mass flow rate from the ball mill. Of course it may be appropriate to include suitable lag periods in the calculations in order to compensate for the circuit holdups. Although density guages and magnetic flowmeters are expensive and require sustained maintenance and calibration, they do provide for an accurate estimate of the circulating load.

From Figure 5.3 it is apparent that the overall controller performance is significantly better than any of the previous configurations examined. The COS control is extremely good, and it is felt that the 3 tph deviation in CL is not that significant. The final steady state FOF is about 45 tph which is comparable to the highest value obtained previously, see Figures 4.7 and 4.8. From the tabulated stream synopsis available from the simulator, or by performing the two steady state mass balances, it is apparent that the product stream density drops from just over 52% at the onset of the hardness change, to just under 40% at the new steady state values. This may be of some concern in those operations in which flotation performance deteriorates if the feed density falls below a certain level. Once again, this drop in product stream density is an important point to note in the controller evaluation, but at this point impossible to quantify in terms of building in controller or input constraints. Such site specific applications are left to

those users with a particular operation in mind.

In order to test the hypotheses that the Type II configuration provided the best pairing of inputs and outputs, an attempt was made to tune the Type I configuration and compare the best performance of it with that presented in Figure 5.3. The tuning problem here was exacerbated by the interactions, as expected, which dictated that very low gains be used on the SWF-CL loop. The best performance achieved, with FOF controlling COS (K_c=9, τ_I=7) and SWF controlling CL (K_c=1, τ_I=45), yielded IAEs of 20.9 and 538 respectively. Figure 5.4 indicates the controller performance, and comparison with Figure 5.3 reveals two major differences. The first is in the degradation of the CL control, the IAE of which could not be lowered much beyond 500 without experiencing severe loss in the COS control and ensuing stability problems. The interaction problems are clearly seen in the oscillations of both the inputs and the outputs, although the final steady state values are quite close to those seen in Figure 5.3. Although the COS controller gains are quite similar to the Type II COS gains, the CL gains are seen to be significantly smaller, indicative of the detuning required due to the interaction problem. The obvious conclusion at this point is that the Type II controller out performs the Type I controller significantly, which in turn validates the results obtained from the relative gain array.

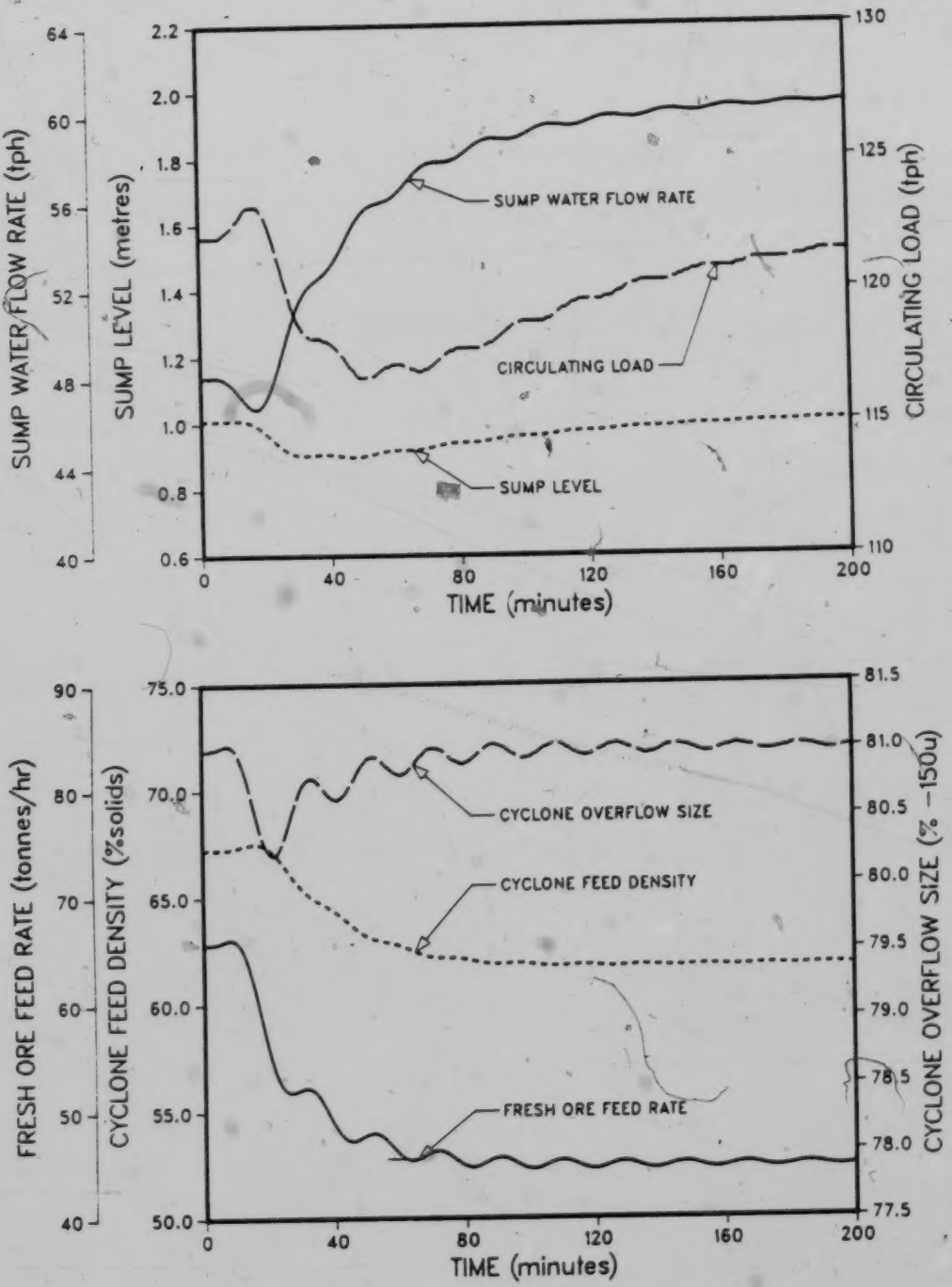


Figure 5.4 Type I Control, Response to a Step Increase in Ore Hardness at t=2 Minutes

5.3 Evaluating Type II Control

Up until this point all control work has only considered a single disturbance, i.e. a step increase in ore hardness. In highly non-linear systems it is quite common to find that a tuned controller that can adequately handle a positive disturbance has to be re-tuned in order to control the equivalent negative disturbance. Thus the next logical step is to examine the controller as the ore hardness increases, then when a new steady state has been achieved, have the ore hardness decrease and let the simulator come to a new steady state. This is simulated quite easily by simply choosing a square wave hardness disturbance and setting the period to 400 minutes so that the changes occur every 200 minutes. Of course it is only necessary to run the simulator for 400 minutes in order to fully evaluate the controller performance. Figure 5.5 indicates how well the controller works, with controller gains being set equal to those discussed previously. The respective IAEs were shown to be simply double those recorded for the 200 minute simulations, indicating that the controller performed equally well for the two disturbances.

Other disturbances felt to be important to the grinding circuit performance are extraneous water flowing into the sump and changes in the feed stream size distribution. Of the two, it is the former that can pose the greatest problem. Sump water flow rate disturbances are simulated by combining two water streams, the manipulated water flow rate

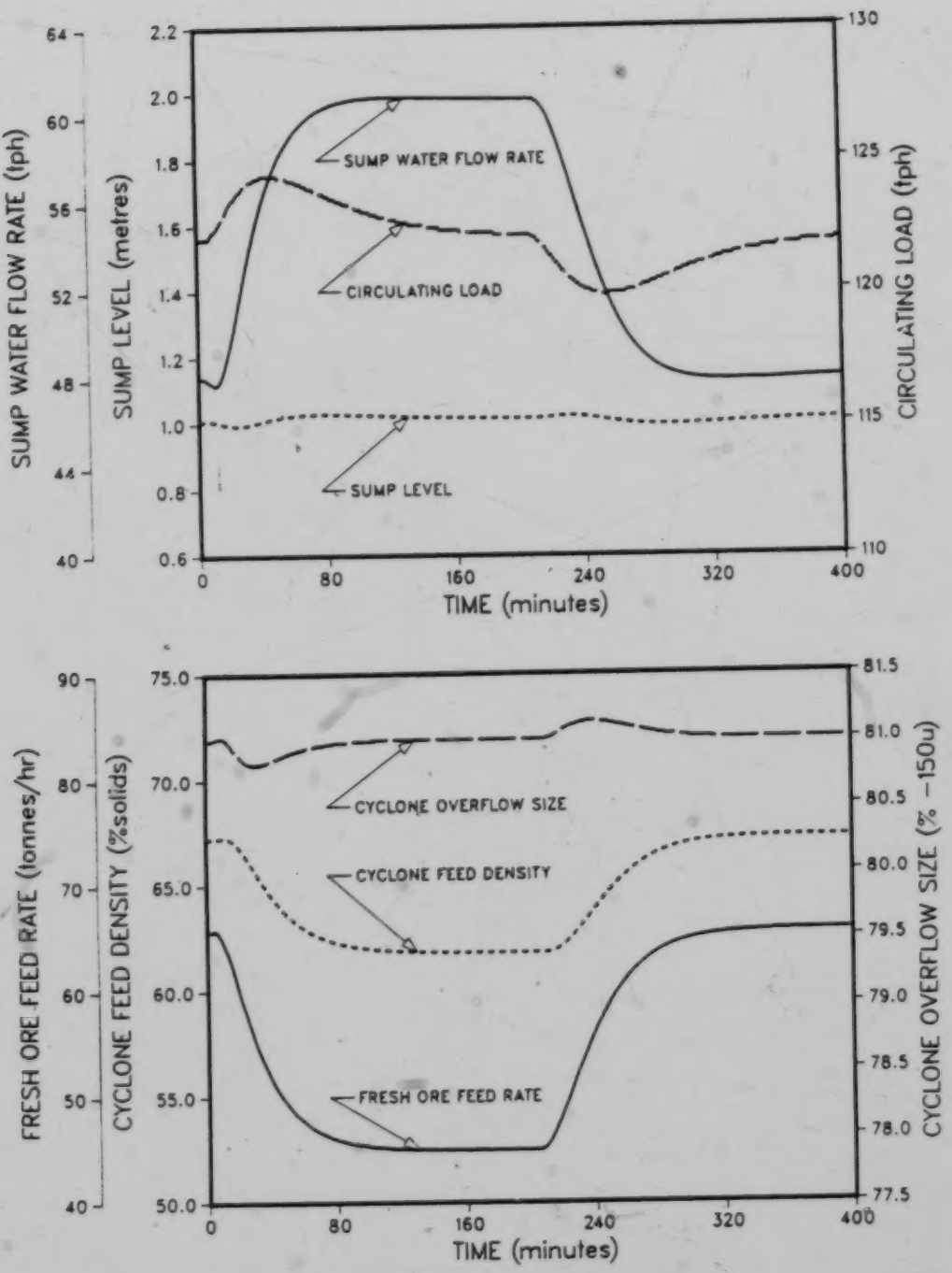


Figure 5.5 Type II Control, Step Increase in Ore Hardness at t=2 Minutes, Step Decrease in Ore Hardness at t=202 Minutes

and the extraneous water flow rate, into one stream that feeds the sump, with this combined stream being monitored in the tabulated output. As such the disturbance algorithm can modify the extraneous water flow rate independently of the flow rate through the manipulated valve. For the purposes of the simulation an extraneous flow of 9.6 tph (approximately 20% of the steady state flow to the sump) was included for a period of 100 minutes. This is felt to be a realistic reflection of what could possibly occur in the plant, e.g. operators hosing down the floor after a spill, undetected leakage of a water pipe in the plant, or a malfunctioning valve on the manipulated sump water line. Figure 5.6 indicates the efforts of the controllers to maintain the COS and CL set-points in the face of the extraneous water disturbance. There is some deterioration in the COS control (IAE=11.6), but this has to be expected given the nature of the relationship between COS and SWF. The CL control is excellent (IAE=29), indicative of the speed at which the controllers adjust the inputs. The sump water flow rate curve shows that the controller acts to reduce the manipulated flow rate such that the total flow to the sump equals the steady state flow rate observed prior to the onset of the disturbance. Since in virtually all industrial situations it is impossible to measure the extraneous flow to the sump, thus precluding any form of feedforward control, controller performance such as this would certainly be deemed to be quite adequate. A note of caution here is that it is quite

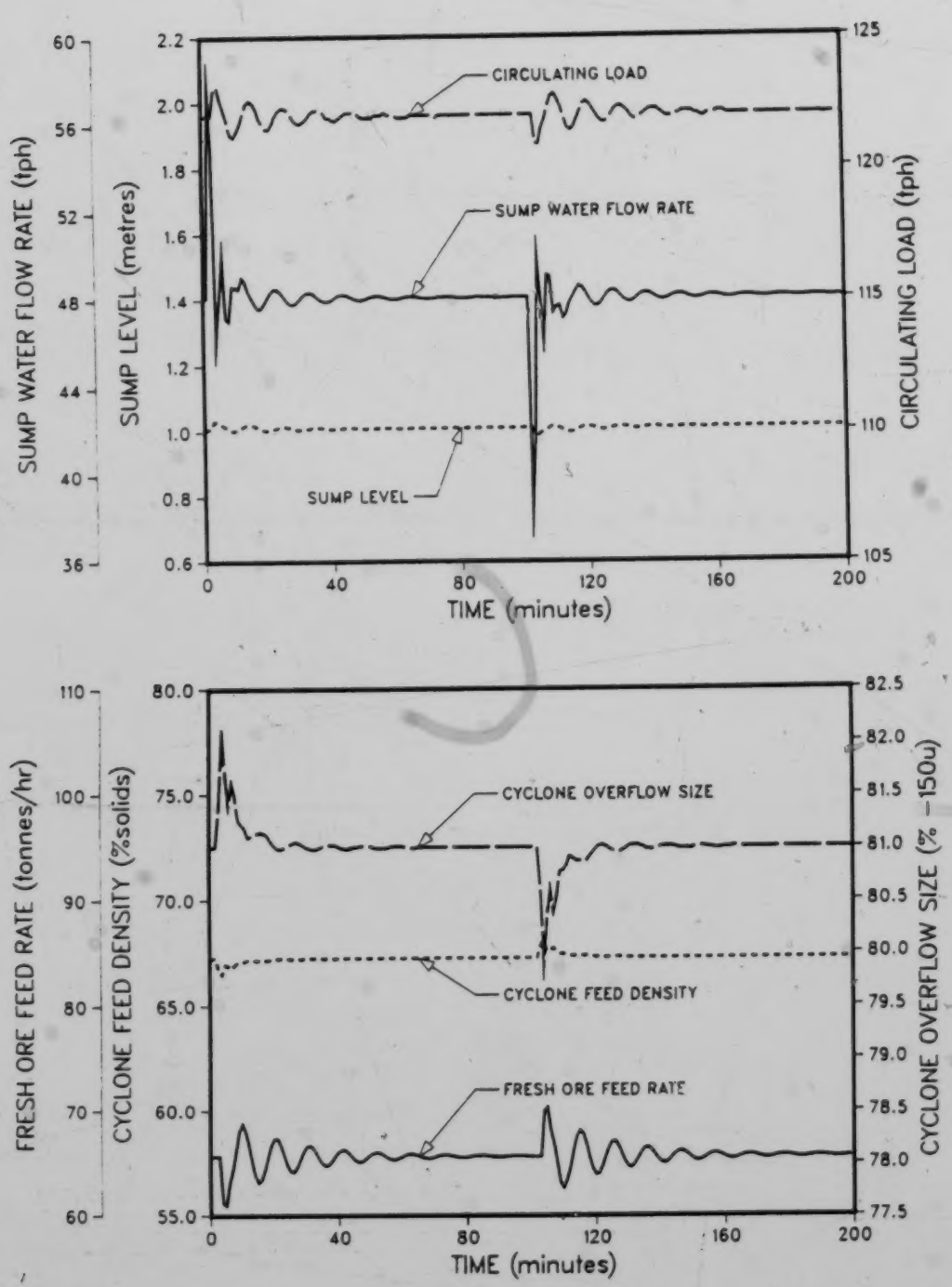


Figure 5.6 Type II Control, Extraneous Water Flow to Sump - 9.6 tph for 100 minutes, starting at t=2 minutes

easy to simulate extraneous sump water disturbances that severely degrade the controller performance, or cause the system to oscillate. However, it is felt that these situations are due to unrealistic disturbances, either the period of the disturbance is too small or the amplitude too great, such that they are not a true reflection of what could occur in the plant. If such wild fluctuations are experienced in the operating environment then it is not so much a control problem as one of poor operator performance or training, inadequate maintenance procedures or possibly inappropriate circuit design, all of which may require some form of upgrading before the control scheme will exhibit its full effectiveness.

The last circuit disturbance to be studied is a change in the rod mill feed size distribution. From Figure 3.19 it is apparent that an increase in the feed coarseness has much less of an effect on circuit performance than the other disturbances considered. Figure 5.7 clearly indicates that an increase in the feed coarseness is easily handled by the controllers, resulting in a slight decrease in FOF and a similar increase in SWF in order to maintain the set-points. Due to the fairly limited fluctuations in the rod mill feed size distribution, it is felt that a well tuned Type II controller, designed specifically to reject hardness disturbances, should be quite capable of handling this type of disturbance.

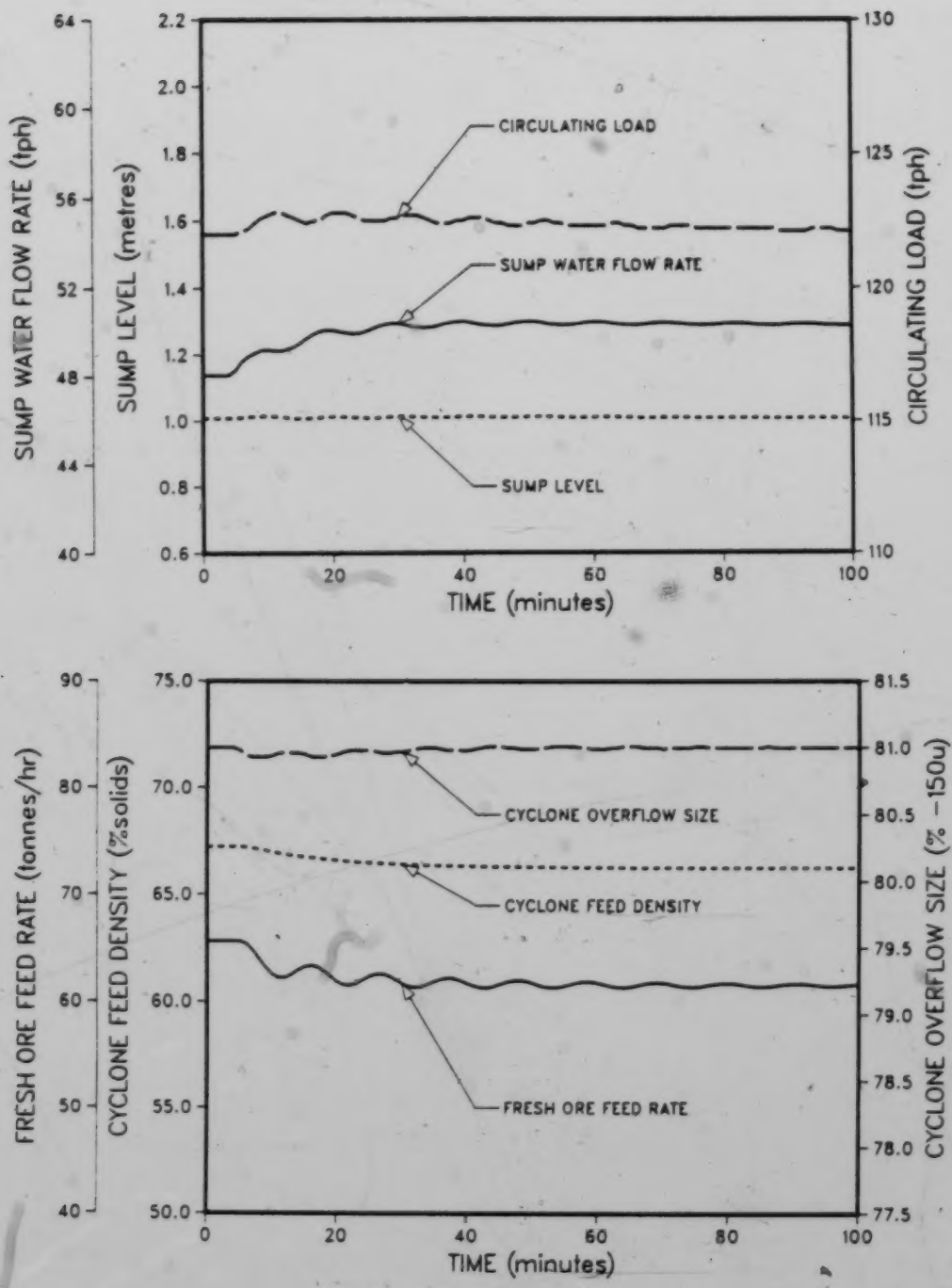


Figure 5.7 Type II Control, Step Increase in Rod Mill Feed Coarseness at t=2 minutes

5.4 Servo Control

Although not felt to be of primary concern in this particular work, it is of interest to evaluate the controller performance in the presence of set-point changes. Flintoff and Wood (1985) and McDougall (1987) indicated that for the controllers they studied, notably FOF controlling COS, some retuning was required when changing from regulatory control to servo control. Although not an integral part of the simulator, set-point changes can be made by simply adding a few lines of code to the driver since control is returned to the driver every iteration. Figures 5.8 and 5.9 indicate the controller performance for servo control of COS and CL respectively. As with the regulatory control it is quite easy to degrade the controller performance by simply demanding unrealistic changes. In both cases the set-point changes have to be reasonable, and enough time allowed between changes for the system to attain a new steady state. In the industrial situation, any set-point changes are usually ramped in to avoid sudden upsets in the circuit. Step changes in the set-points are evaluated here simply as a method of testing the robustness of the controllers and for the sake of programming ease. From Figure 5.8 it is apparent that a 1% increase followed by a 2% decrease in the COS set-point is adequately handled by the controllers, although it would appear that a 2% change in COS set-point is approaching the upper limit of a suitable step change. The sump water flow rate curve in Figure 5.8 is a bit misleading

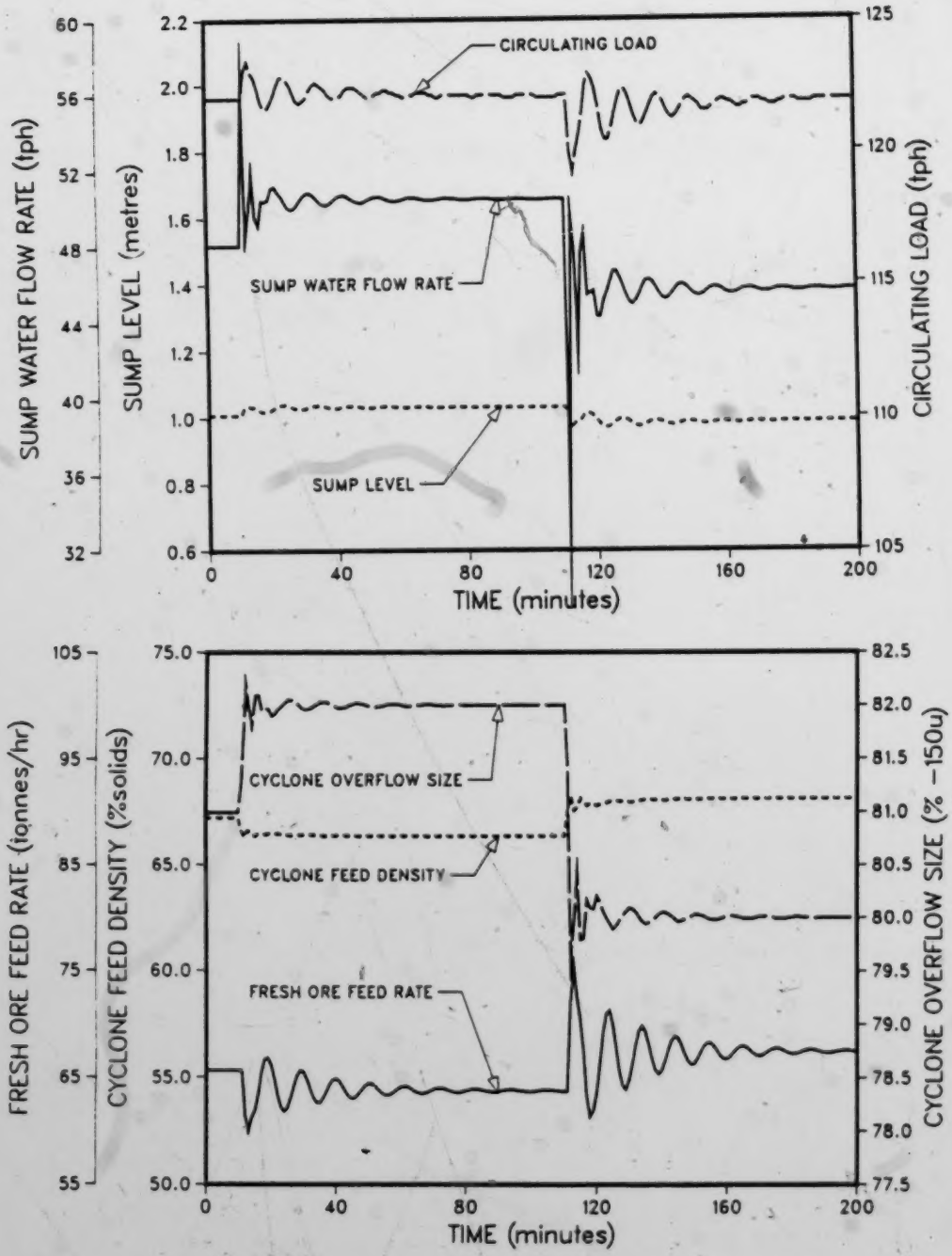


Figure 5.8 Servo Control, Controller Response to COS Set-Point Changes

as it crosses the ordinate at about t=110 minutes. The tabulated value at this point is actually 33.58 tph and the crossing over of the axis is due to the action of the plotting package used to generate this figure. The overall performance is quite good with IAEs of 8.7 and 58 for the COS and CL respectively.

Figure 5.9 shows the circuit response to subsequent increases in the circulating load set-point of 4 tph. Once again the magnitude of these changes have to be kept realistic in order for the controllers to performa adequately. The IAEs of 9.1 and 58 for COS and CL respectively are still quite acceptable, but the oscillations on the FOF input are starting to become substantial, probably indicating that a 4 tph set-point change is about as large as the system can handle. The overall performance of both servo controllers could probably be further improved by ramping in the set-point change rather than implementing a step change in the set-point.

The important point to make in regards to this particular investigation is that no retuning of the controller gains was required in order to implement the servo control. This is not to say that some improvement in performance couldn't be achieved through further tuning, but it is reassuring to note that the controller functions adequately in both the regulatory and servo control mode. In the industrial situation, changing the control parameters as a function of the mode of control wouldn't appear to be conducive to

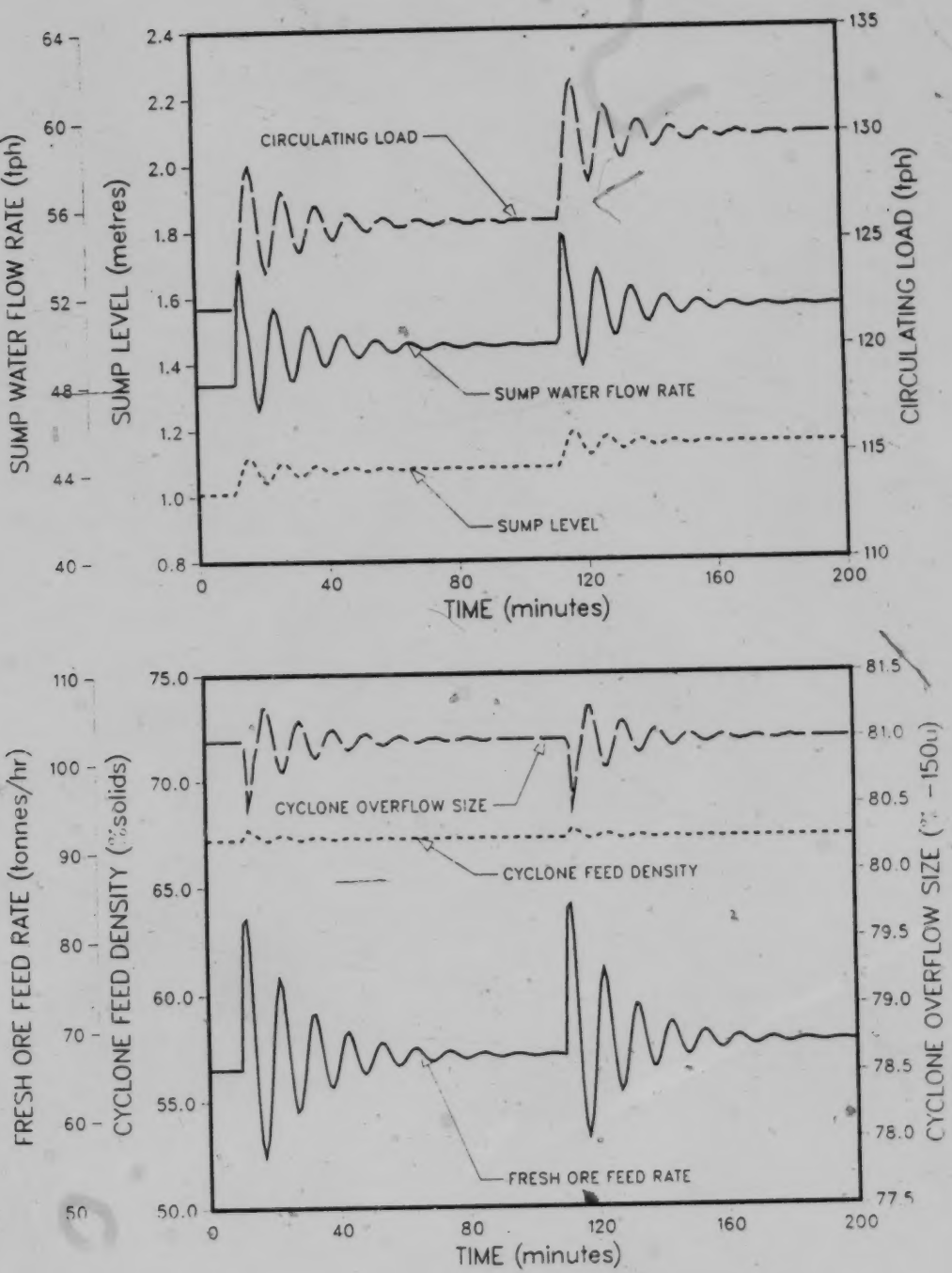
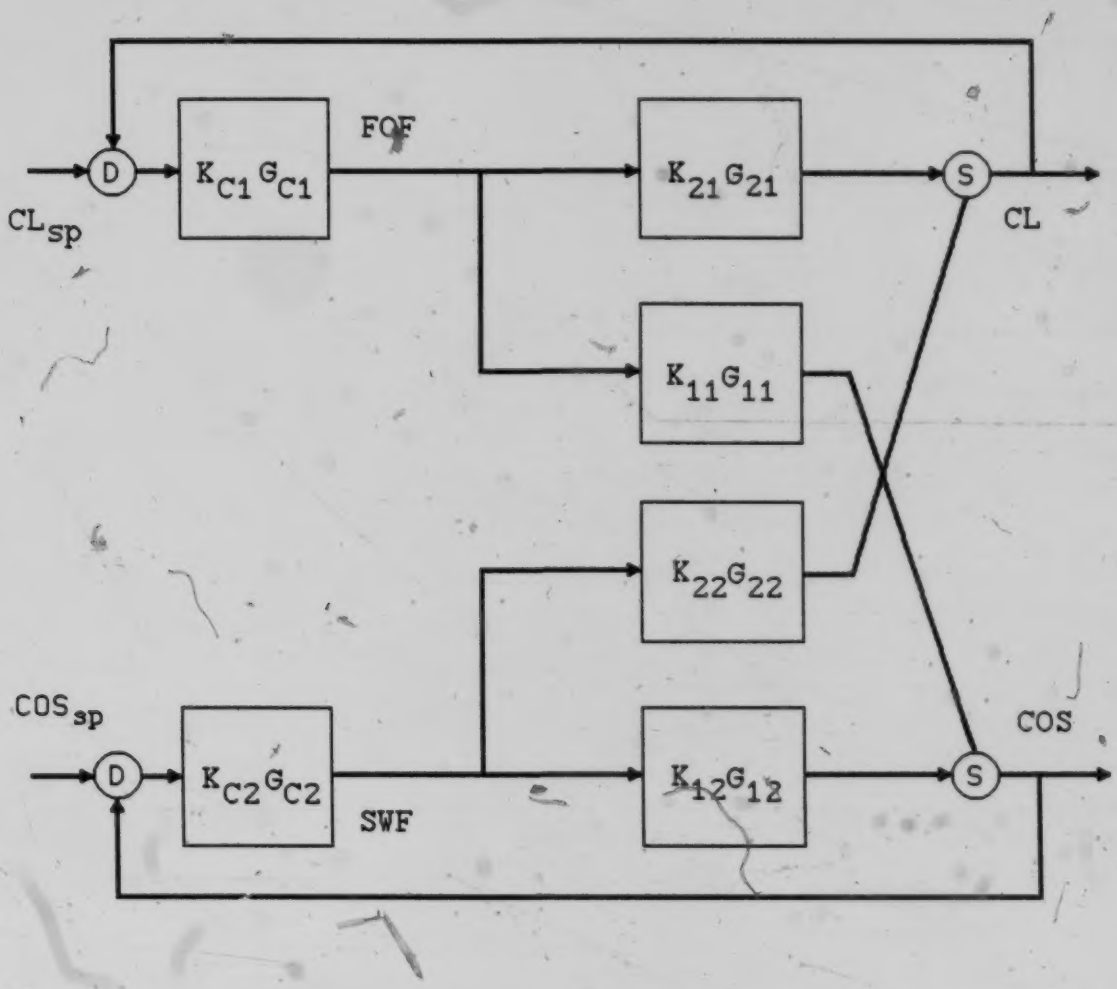


Figure 5.9 Servo Control, Controller Response to CL Set-Point Changes

good overall control. Why other workers found that retuning was required, while this author with a slightly different simulator did not, is not immediately apparent. One possible answer could lie in the comparison of the controller gains used in the various works. In the current work, the control action is much slower, lower K_c and larger τ_I, which it is felt, more realistically reflects the large lag periods in the circuit. As such, this seems to imply more robust control, i.e. adequate rejection of various types of disturbances, with the added bonus that the controllers can also deal with set-point changes.

5.5 Type II Control with Static Decouplers

It would appear that the major limitation to further improvement in the controller performance is hindered by the degree of interaction between the two loops. This was quite apparent throughout the tuning process in that any attempts to significantly tighten the control of one of the loops resulted in severe deterioration in the performance of the second loop. The reasons for this can be identified by studying the block diagram representation of the grinding circuit as shown in Figure 5.10. The numbering of the process transfer functions are such, so as to remain consistent with the relationships derived in the earlier section on interaction analysis. From Figure 5.10 it is obvious that a change in either of the inputs has an effect on both of the outputs via the appropriate transfer functions. Thus



LEGEND:

K_{ij} - Controller or Process Transfer Function Static Gain
G_{ij} - Controller or Process Transfer Function Dynamic Component
CL - Circulating Load
COS - Cyclone Overflow Size Distribution
FOF - Fresh Ore Feed Rate
SWF - Sump Water Flow Rate
sp - Set-point
(S) - Summation
(D) - Difference

Figure 5.10 Block Diagram Representation of the Closed-Loop Grinding Circuit

although the loops have been arranged in order to minimize the degree of interaction, the two control loops still affect each others performance. In certain situations it is possible to improve the controller performance through the addition of decouplers. A decoupler is designed to cancel the interaction effects between the two loops, and in theory if the process models are known well enough, can produce two non-interacting control loops. In the current situation the process models are not fully understood, the process reaction curves are not adequately described by simple first order plus dead time models, and no effort has been made to model them as second order plus dead time processes. However, the process reaction curves do provide a good estimate of the steady state process gains, such that it is possible to design static decouplers which should facilitate further improvement in the controller performance.

The requirement of the decoupler is to cancel out the interaction effect of a change in one of the inputs by manipulation of the second input. From Figure 5.10 it is evident that CL is a function of FOF and SWF, which neglecting the process dynamics, yields

$$CL = K_{21}\ FOF + K_{22}\ SWF \tag{5.12}$$

and similarly for COS

$$COS = K_{11}\ FOF + K_{12}\ SWF \tag{5.13}$$

From Equations 5,12 and 5.13 and Figure 5.10 the objective of the decoupler can be defined so as to yield

$$\frac{\partial CL}{\partial SWF} = \frac{\partial COS}{\partial FOF} = 0 \tag{5.14}$$

From 5.12 and 5.14

$$\frac{\partial CL}{\partial SWF} = 0 = K_{21} \frac{\partial FOF}{\partial SWF} + K_{22} \tag{5.15}$$

or

$$\frac{\partial FOF}{\partial SWF} = \frac{-K_{22}}{K_{21}} \tag{5.16}$$

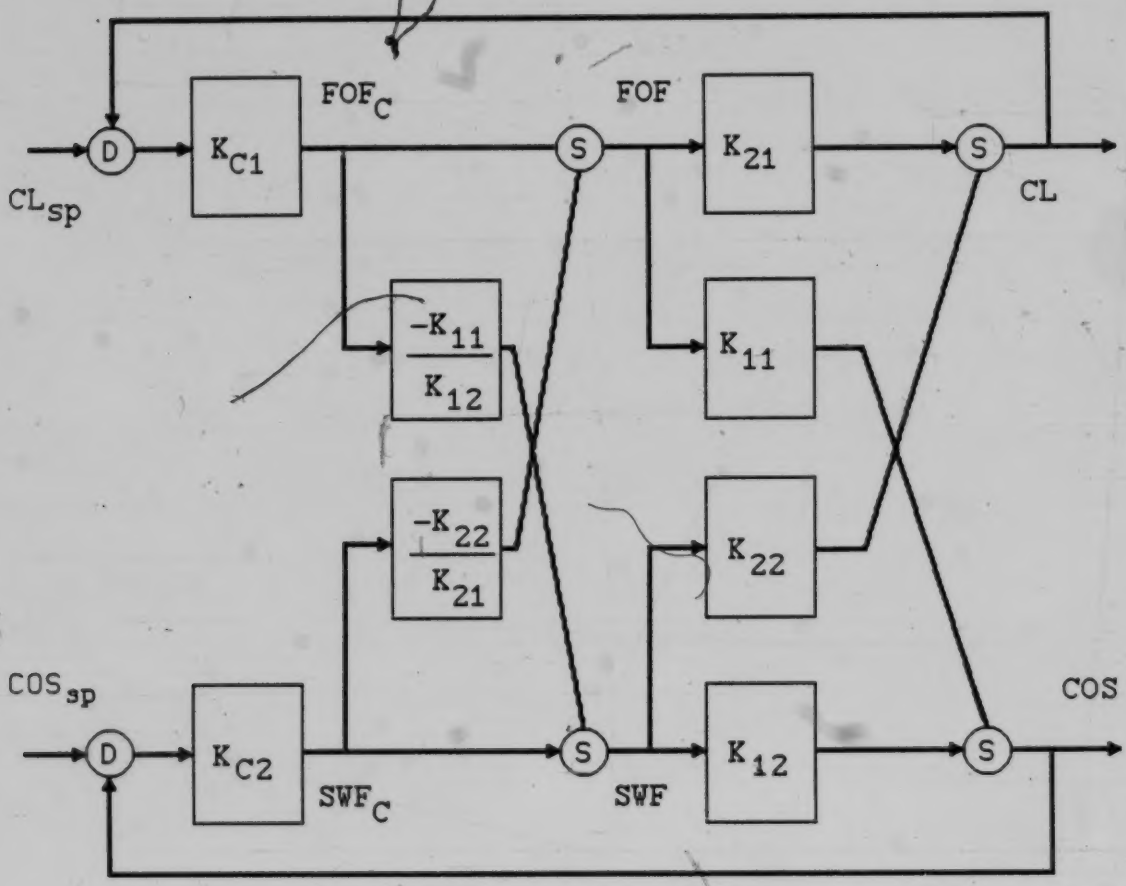
And by similar arguments for 5.13 and 5.14

$$\frac{\partial SWF}{\partial FOF} = \frac{-K_{11}}{K_{12}} \tag{5.17}$$

It is Equations 5.16 and 5.17 that define the form of the two decouplers, see Figure 5.11 for the appropriate block diagram. To implement the decoupler it is necessary to define two new manipulated controller outputs, FOF_C and SWF_C, which are related to the decoupled system inputs as follows

$$FOF = FOF_C - \frac{K_{22}}{K_{21}} SWF_C \tag{5.18}$$

$$SWF = SWF_C - \frac{K_{22}}{K_{21}} FOF_C \tag{5.19}$$



LEGEND:

K_{ij} - Controller or Process Transfer Function Steady State Gain
CL - Circulating Load
COS - Cyclone Overflow Size Distribution
sp - Set-Point
FOF - Fresh Ore Feed Rate
SWF - Sump Water Flow Rate

Subscript "C" indicates input due to the controller output

(S) - Summation

(D) - Difference

Figure 5.11 Block Diagram of Grinding Circuit with Static Decouplers Included

Since the K_{ij} values and the decoupled system inputs are known it is possible to back calculate FOF_C and SWF_C. Using the static gains calculated previously in this chapter and the steady state values for the system inputs, the new controller outputs can be calculated as

$$FOF_C = 57.369 \text{ tph} \tag{5.20}$$

$$SWF_C = -12.469 \text{ tph} \tag{5.21}$$

Given that the new sump water flow rate controller output is a negative value results in some modification to the DYFLO PI algorithm since it constrains the output between 0 and 100 to simulate the normalized output signal. A slightly modified version was develop which still constrains the FOF_C output between 0 and 100, but allows the SWF_C output to range between -20 and 80. This may not be absolutely correct from a theoretical point of view, but allows the control action to be implemented with a bumpless transfer. Since the process inputs are now a function of both controller outputs, they are constrained to the ranges used in previous work, i.e. 20-120 tph for FOF and 0-100 tph for SWF.

To test the effectiveness of the two decouplers it is only necessary to run another pair of open loop simulations. Figure 5.12 shows the result of a 10 tph increase in SWF. Two important points to note are, first, the compensatory decrease in FOF, and secondly, the relatively small change

in CL, i.e. the output decoupled from the SWF input. This indicates that the decouplers are having some positive effect on the performance of the circuit. Similarly, for Figure 5.13, simulating a 10 tph decrease in FOF, the same type of behaviour is observed. There is an immediate adjustment of SWF, and although there is a significant amount of movement in COS, the final steady state is very close to the starting steady state, i.e. the relative gain is approximately zero. The movement in the COS output is due to the step change in SWF and the fact that the SWF/COS dynamics are much faster than the FOF/COS dynamics, such that the compensating reduction in SWF affects COS before the result of the change in FOF is felt in the circuit.

Tuning the decoupled system followed the same procedure as outlined before, with initial controller parameters being derived using the empirical techniques of Cohen and Coon. Interestingly enough, these estimates produced good control if only one of the loops was being controlled. When both loops were being controlled simultaneously using the Cohen and Coon estimates the performance was very poor and highly oscillatory. This sort of behaviour is indicative of the discrepancy in the dynamics of the two loops. With severe detuning of the FOF/CL loop, controller gains were derived that resulted in performance in the presence of a hardness change that was significantly better than that observed previously, see Figure 5.14. For the SWF/COS loop (K_c=7.5, τ_I=1.6) and for the FOF/CL loop (K_c=2, τ_I=40) the IAEs are

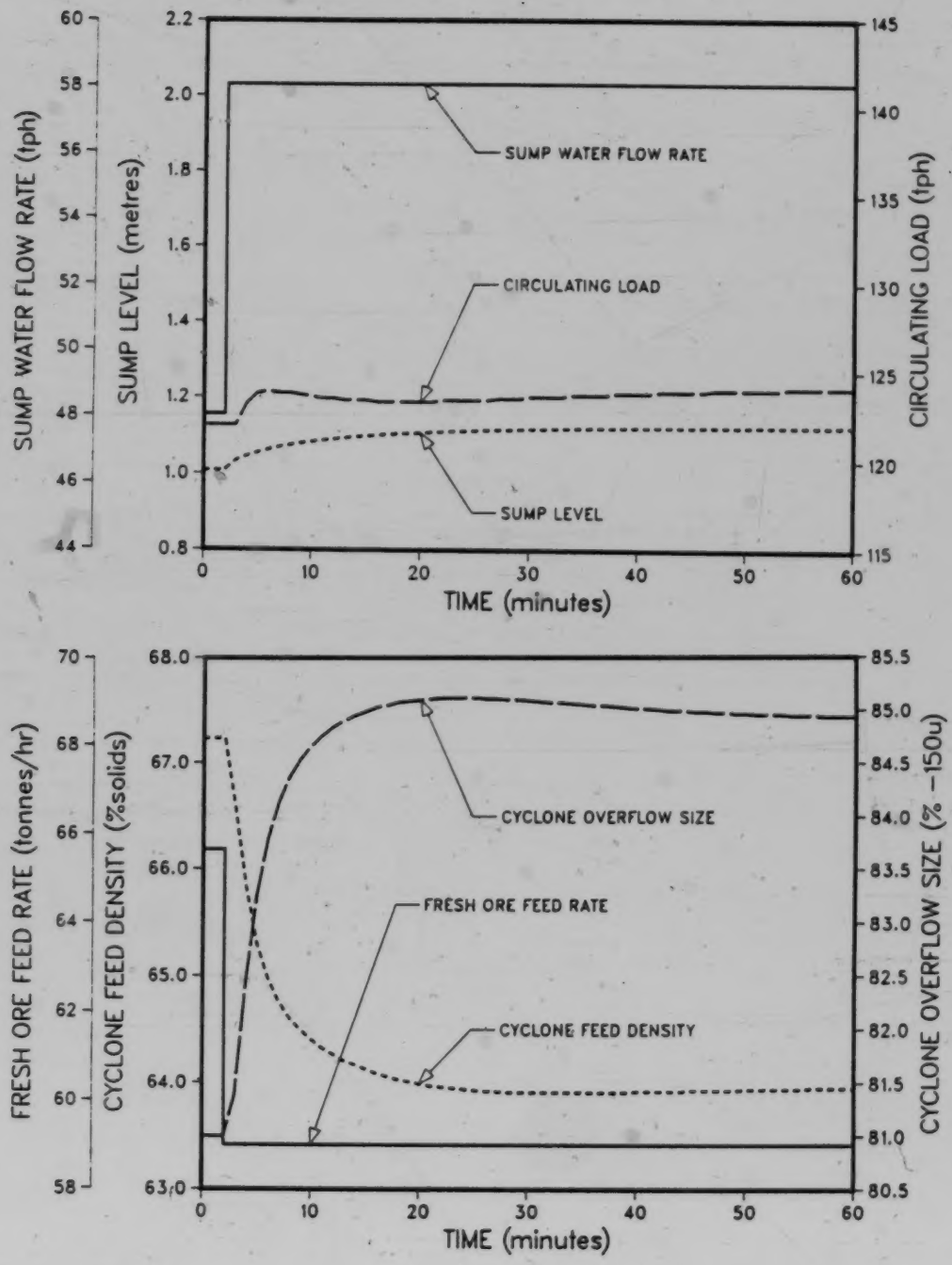


Figure 5.12 Open-Loop Response to a 10 tph Increase in SWF using Static Decouplers

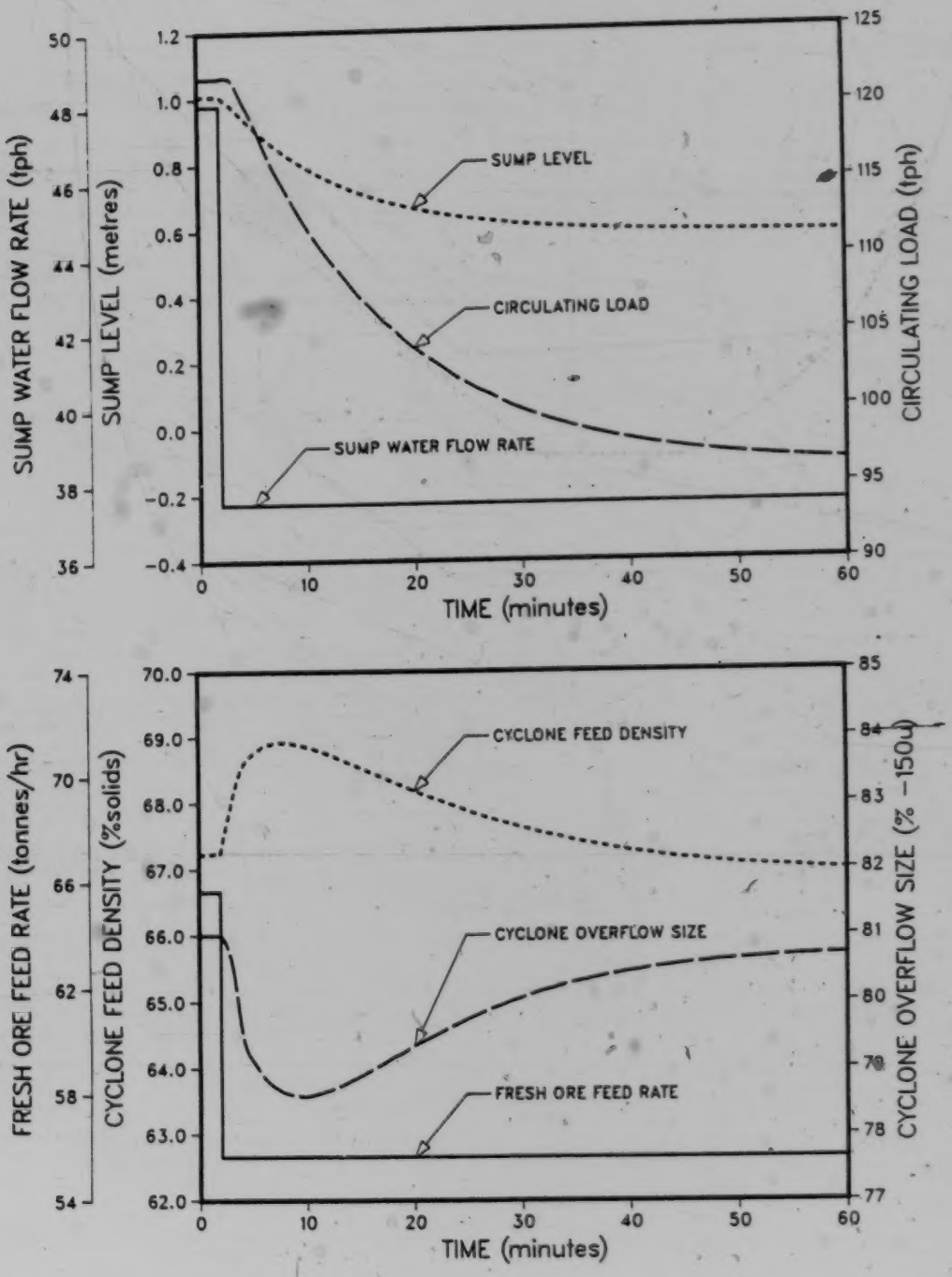


Figure 5.13 Open-Loop Response to a 10 tph Decrease in FOF using Static Decouplers

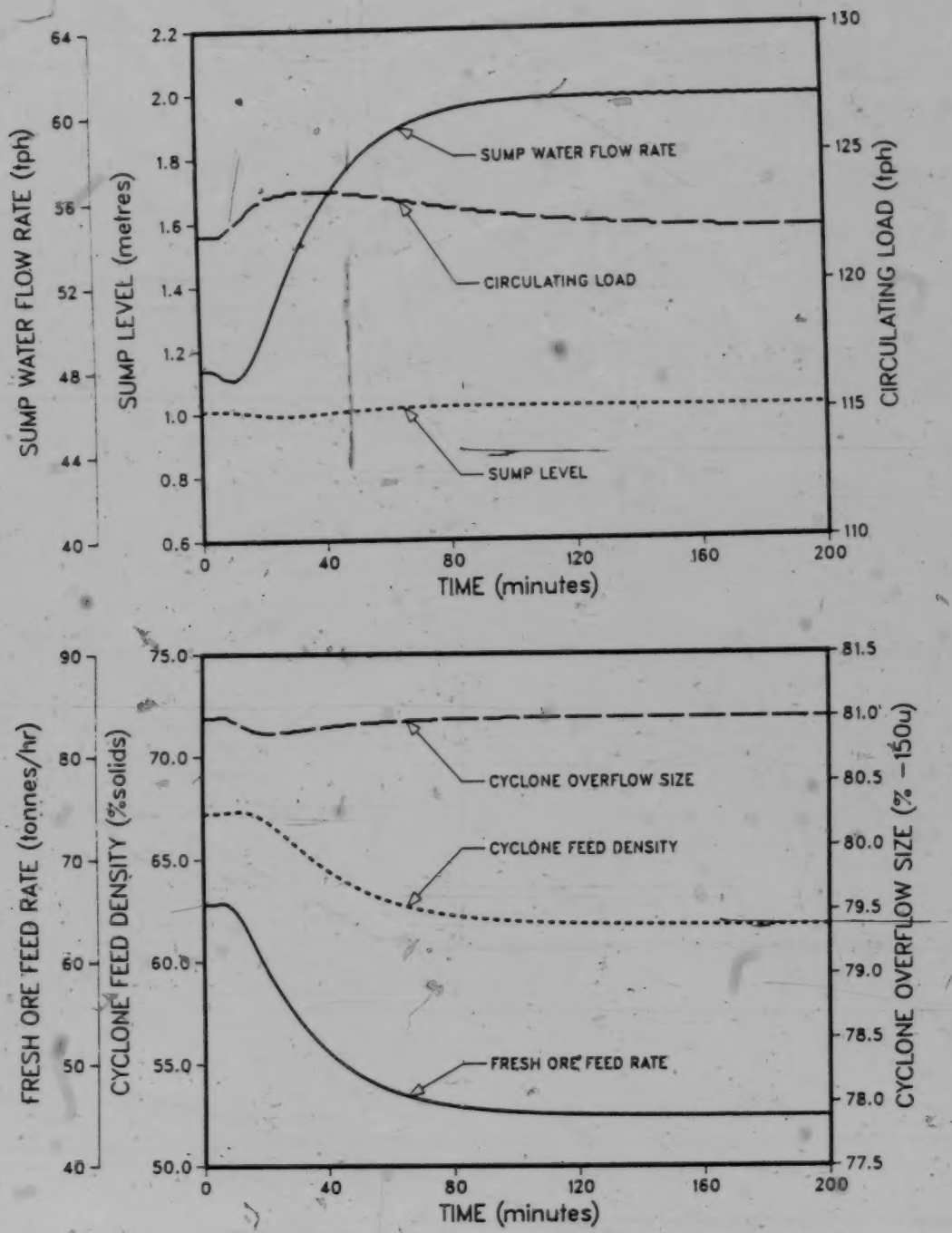


Figure 5.14 Type II Control with Static Decouplers, Response to a Step Increase in Ore Hardness at t=2 Minutes

4.52 and 142 respectively. This corresponds to a 40% reduction in the COS IAE and a 30% reduction in the CL IAE. How this improvement in overall performance relates to the plant economics is impossible to say, but undoubtably the improvements are significant, and the cost of implementing static decouplers once a control system is in place is minimal.

For reasons of completeness it is necessary to investigate the system linearity by controlling both a step increase and a step decrease in the ore hardness. The results are presented in Figure 5.15, and once again the IAEs are simply double those recorded in the previous simulation. Although other regulatory and servo control simulations have not been done, it is fully expected that the decoupled controller performance will be better than that observed for the simple Type II controller.

5.6 Type II Control with Static Decouplers Using a Fixed Speed Cyclone Feed Pump

At this stage of the work it is quite easy to generate a lot of data in the course of studying various embellishments to the work already presented. Of all the questions that still require answering, it is one involving the use of a variable speed pump that is possibly the most pressing. In the initial stages of developing a grinding circuit control strategy it was noted that a fixed speed pump had severe limitations when FOF was used to control COS in that the reduced feed rate resulted in the sump running dry. The

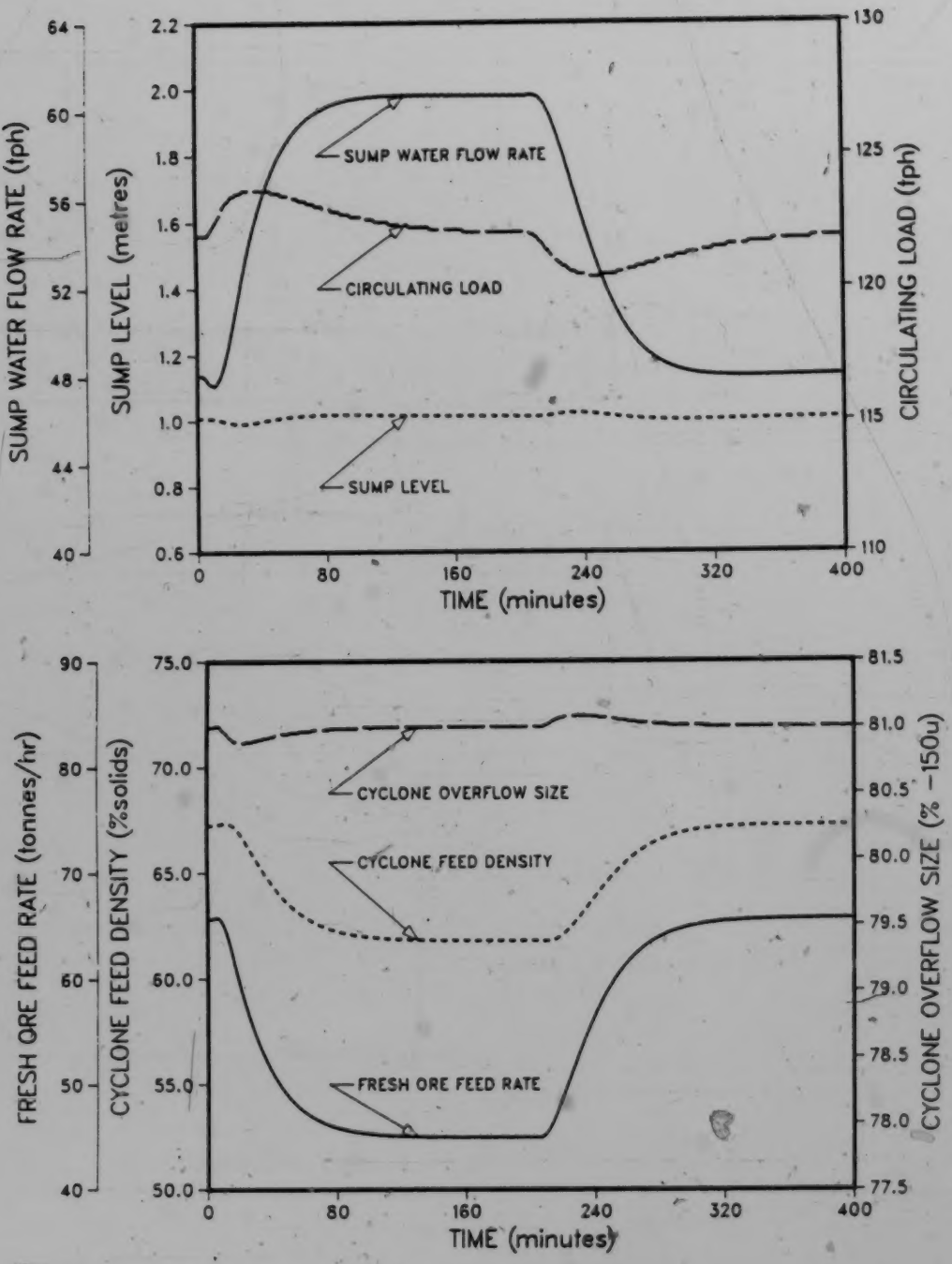


Figure 5.15 Type II Control with Static Decouplers, Response to a Step Increase in Ore Hardness at t=2 Minutes, Step Decrease in Ore Hardness at t=202 Minutes

argument that using a variable speed pump to control sump level created a third input so that Type I and Type II control strategies could be investigated was used to justify its inclusion. Implementation of the Type II controller, using a variable speed pump, resulted in good circuit performance in the presence of various disturbances. An important point that was never raised in the discussion of the circuit behaviour was the lack of movement in the sump level even while the two circuit inputs were being manipulated to implement control. Since the level controller is proportional, one would expect to see some offset due to the control action. Very little movement in the sump level could possibly indicate the redundancy of the controller. To test this hypotheses a simulation was run using a fixed speed pump and the decoupled Type II controller. The controller parameters were simply those used in the previous section and the simulation results are presented in Figure 5.16. The COS and CL IAEs are 4.4 and 143 respectively, values almost identical to those recorded in the simulation that generated the data for Figure 5.14. The only significant difference between Figures 5.14 and 5.16 is the behaviour of the sump level. However, the fluctuations observed when the fixed speed pump is used are tolerable when one considers the total available depth of the sump. Once again, in order to make the presentation complete, a full 400 minute simulation is included to show the performance of the controller when both a step increase and a step decrease in the ore hardness is

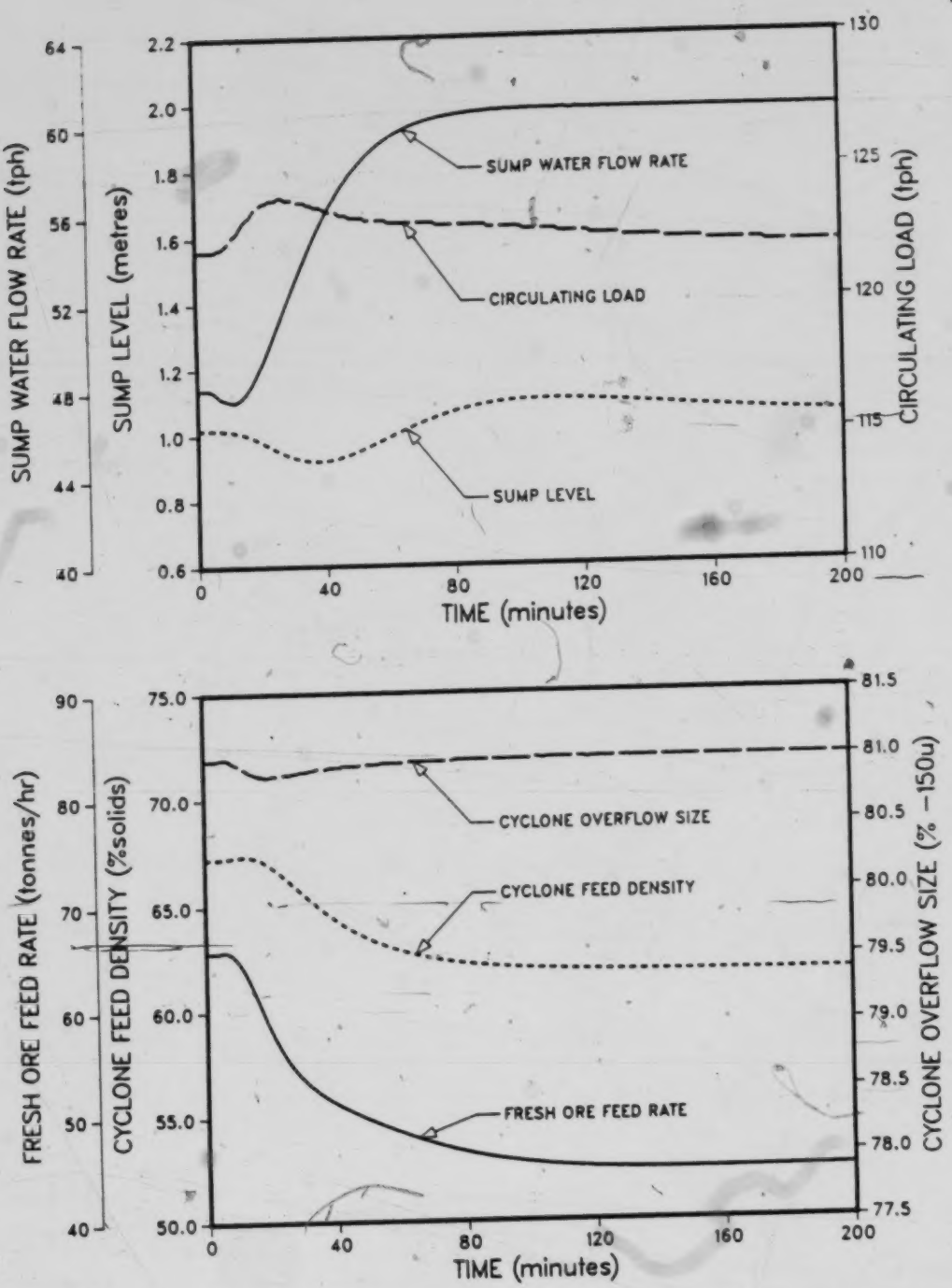


Figure 5.16 Type II Control, Static Decouplers, Fixed Speed Pump, Response to a Step Increase in Ore Hardness at t=2 Minutes

experienced.

These results have some important repurcussions. Currently in Canada very few mineral processing plants have variable speed cyclone feed pumps. One installation that the author is familiar with, a northern Canadian gold operation, does have a variable speed pump, but the circuit configuration is such that it has to be operated close to its top speed at all times, thus in essence making it a fixed speed pump. In addition only a very few installations in the United States, Australia or South Africa indicate that a variable speed pump is currently in use (e.g. Gossman et al., 1982). From this current work it would seem that adequate Type II control of a grinding circuit can be achieved without going to the expense of replacing a fixed speed pump with the more expensive variable speed pump. Of course, it is impossible to say at this moment whether or not this may simply be a coincidence of this particular circuit, i.e. the volumetric increase in the sump water flow rate just happens to offset the corresponding decrease in that of the fresh ore feed rate. However these results do seem to contradict the work of other authorities, Gault et al. (1979) and Lynch (1977), who state that effective control of short term fluctuations in grinding circuits is not possible with a fixed speed pump.

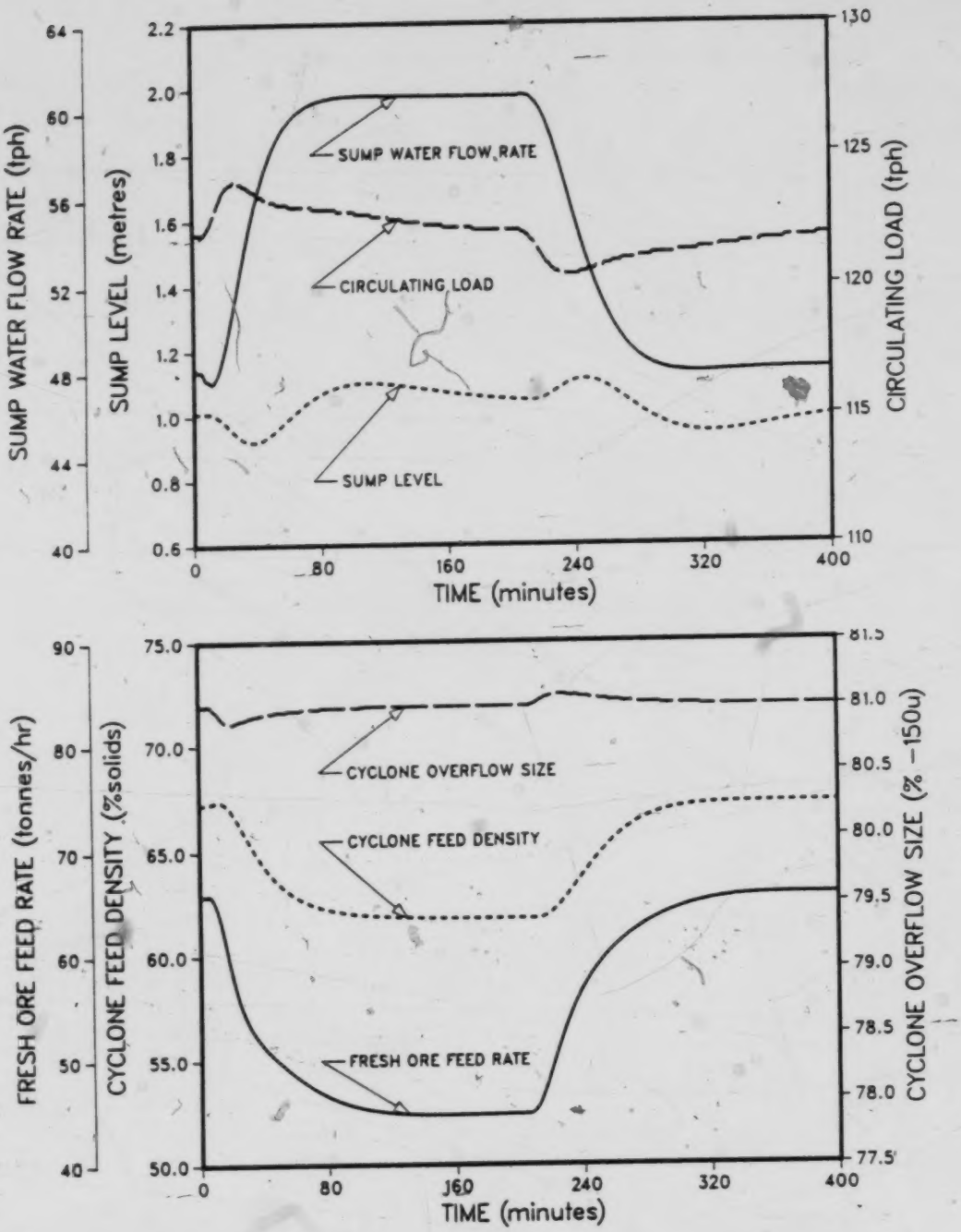


Figure 5.17 Type II Control, Static Decouplers, Fixed Speed Pump, Response to a Step Increase in Ore Hardness at t=2 Minutes, Step Decrease at t=202 Minutes

5.7 Summary

Synthesis and evaluation of the relative gain array indicated for the circuit at hand that a Type II control system yielded the preferred pairing of control loops. These results were validated through simulations which indicated that the interactions experienced in the implementation of Type I control were too severe to realize adequate controller performance. The performance of the Type II controller was shown to be significantly better than any of the configurations tested in the previous chapter. Since it was realized that loop interactions precluded further improvements in the controller performance, the inclusion of static decouplers in the control system was studied. With a little extra effort, and some slight modification of the DYFLO PI control algorithm, a Type II controller with static decouplers was implemented. The improvement in performance, although difficult to quantify at the plant level, was felt to be significant, and certainly justified the extra effort. Since it was noted that the sump level showed very little movement with this type of control scheme, simulations were run using the Type II controller with static decouplers and a fixed speed pump. The overall performance was found to be as good as that experienced when sump level was controlled by manipulating pump speed. Whether or not this is simply a coincidence of the system at hand cannot be determined at this time, but it certainly raises a few questions for those individuals who maintain that good grinding circuit control

cannot be achieved without the use of a variable speed pump.

6. Conclusions and Recommendations

6.1 Conclusions

A review of the status of dynamic simulation in the mineral processing industry indicated a real need for a general purpose, interactive, dynamic simulator for an industrial scale rod mill-ball mill grinding circuit. The culmination of work by various individuals over the past few years at the University of Alberta has resulted in just such a package. Currently two versions of the simulator exist, one running of the University of Alberta mainframe under the FORTRANVS compiler (FORTRAN 77), the second running on the IBM PC or XT using the DOS 3.2 operating system and the MS FORTRAN compiler. Currently three different grinding circuit configurations can be simulated in the open-loop mode, open circuit, conventional closed circuit and reversed closed circuit. However, to date, by virtue of its overwhelming popularity in industry, it is only the reversed closed circuit that has been studied for control purposes. A fully interactive front end has been developed in order to avoid the time consuming job of building cumbersome data files prior to using the simulator. Individual simulations are executed by choosing options from a series of menus, which given the fully modular structure of the simulator, can be easily expanded to meet individual user requirements in the future. The simulator developed has been shown to be a powerful tool for examining the dynamics of a grinding circuit under both

open-loop and closed-loop simulations. The final phase of the simulator provides the user with three options to recover the results of each run. Transient information can be saved in one file for future plotting, a stream synopsis is available to indicate the condition of each stream at the end of the simulation, and finally, a complete data output file can be saved so that subsequent simulations can be started from the final conditions of that particular simulation.

The unit models used in the simulator have been shown to be quite adequate for the situation at hand. The form of the mill model used has been verified against pilot plant and industrial scale grinding mills (Rajamani, 1979) and is felt to be as good as any currently available. The grinding mill algorithm can handle a feed stream with one or two solids components and simulate up to three mixers in series. A hydrocyclone model based on the work of Lynch and Rao (Lynch, 1977) and verified in industrial situations (Smith and Guerin, 1980; Draper et al, 1969) is included to simulate the grinding circuit classifier. The models for conveyors, pipes, sump and pumps have been developed at the University of Alberta and are felt to be sufficiently accurate for the simulation studies covered in this work. For users preferring different models, the highly modular structure facilitates substitution of the appropriate algorithm. A separate control block is included to allow the user a choice of control schemes for each simulation. Once

again, the menu driven structure in this area facilitates future expansion.

The grinding circuit is described as a 3x3 system and various open-loop simulations were performed in order to study the system interactions. The nature of the circuit performance to various disturbances indicated that some form of sump level control was required in order to study classical multivariable control schemes. By manipulating pump speed to control sump level it was possible to study the Type I and Type II control systems. Interaction analysis indicated that the Type II system, using fresh ore feed rate to control circulating load and sump water flow rate to control the product stream size distribution, yielded the preferred pairing. These results were borne out through further simulation studies. Although the circuit performance was greatly improved, and the controller found to be effective in both regulatory and servo control, it was recognized that loop interactions hindered any further improvement in performance. As a result, the implementation of static decouplers was studied and found to improve the performance criteria by more than 30%. Since it was noted that sump level remained fairly constant during the simulations, tests were performed to determine the redundancy of the sump level controller. Simulations indicated that for the particular circuit being considered, Type II control with static decouplers was equally effective using a fixed speed pump which results in the sump level fluctuating slightly. This raises

some important questions regarding current grinding circuit control philosophy, which indicates that good control cannot be achieved with a fixed speed pump. Unfortunately it is not possible to settle this question without simulating other circuit configurations, and as of now, the information needed for this is not available.

At the onset of the project it was felt that certain limitations to classical feedback control of grinding circuits would be revealed, and as such, various modern control schemes could be investigated. However, personal communication with individuals well versed in these areas, particularly predictive control, indicated that the feedback control was so good that very little benefit could be gained from implementing these schemes.

6.2 Recommendations

Although the author feels that a complete package has been developed as a result of this work, it is also recognized that the potential for expanding the simulator is limited only by the individual user's ambition. Of primary importance at present is the verification of the simulator against valid grinding circuit performance data. Whether this can be achieved in the near future is uncertain, however it would be very reassuring to determine that the simulator provides an accurate representation of a rod mill-ball mill grinding circuit.

Some areas that the author would like to see studied in the future are as follows:

1. Inclusion of other unit models in the simulator and evaluation of their effects on the simulator performance. Most notably, development of a classifier algorithm using the Plitt models and comparison with the Lynch - Rao models.
2. Some modification of the sump model to more accurately reflect the continuous nature of this unit seems to be appropriate. As indicated by Flintoff et al. (1985) the resistance dynamic head (RDH) component of the pump model was developed from steady state operating data, and as such may have some inherent weaknesses. The development of a more general RDH model may result in further improvements in the strength of the simulator.
3. Currently the breakage and selection elements for the mill models have to be entered individually during the initial start-up stage for each component for both mills. This could be improved by offering a choice of various functional forms such that the user only has to specify the model parameters and the program would calculate the individual elements.
4. From the control standpoint, only continuous classical control has been implemented and evaluated to date. The inclusion of discrete control options, a prelude to studying digital control systems, would enhance the value of

the simulator.

5. The inclusion of process and measurement noise models, and a study of their effects on controller performance, would create a more realistic environment for controller evaluation.
6. Further studies should be implemented to try and determine the limitations of classical control of grinding circuits. Herbst and Rajamani (1979) have consistently indicated that situations can arise in which the classical controllers work against each other and upset the circuit. To date, no situation has been simulated in which such an occurrence has been observed.
7. From a numerical aspect, it has been indicated that the stiff system of equations used in the mill models produces some numerical problems. It would be a useful project to evaluate some of the recently developed integration algorithms designed specifically to handle stiff systems. If succesful, this would permit an increase in the integration step size and a reduction in the time required for the computer to perform a given simulation.
8. In the same vein, an evaluation of other first order equations involved in the simulator may indicate that higher order integration schemes can be implemented to increase in step size. The current limitation to this is of course the mill model, but it may be possible to develop a scheme such that the non-mill model differential

equations are only evaluated every second or fourth iteration, which once again will reduce the simulator execution time.

6.3 Simulator Availability

As indicated in the abstract, the simulator is available at nominal cost to any interested parties. The current philosophy is that the complete source code will be made available on 5¼ inch, double sided, double density disks. The simulator has been run succesfully on an IBM XT with 640K of RAM, using DOS version 3.2, and version 3.3 of the Microsoft FORTRAN77 compiler. The simulator executive file requires approximately 284K of RAM indicating that a machine with 512K (the most common configuration), or 384K (in some earlier models), is required. In addition, it is highly recommended that a math coprocessor be used to speed up the run times. Although the simulator has been tested extensively on both the University of Alberta mainframe and the IBM, there is still the potential for some software bugs to be present. As such, it is fully expected that the user has a good background in FORTRAN programming in order to deal with any problems that may arise. For those individuals interested in this package, further enquiries can be directed to:

Dr. Brian C. Flintoff
Dept. of Mining, Metallurgical and Petroleum Engineering
The University of Alberta
Edmonton, Alberta
T6G 2G6
Tel. (403) 432-3774

7. Bibliography

ACSL - Mitchell and Gauthier, Assoc., Inc., P. O. Box 685, Concord, Mass., 01742.

Adel, G. T. and Sastry, K. V. S., (1982). Design Aspects of a Mineral Processing Simulation Package. Proc. 17th APCOM, J. B. Johnson and R. B. Barnes, editors, SME/AIME Publication, pp 681-692.

Alliston, W. H., Elco, R. A., Desalu, A. and Mutafelija, B. A., (1971). Mathematical Modelling and Training Simulators. DISCOP: IFAC Symposium on Digital Simulation of Continuous Processes. Gyor, Hungary, pp 1-5.

Atkins, J. E., Barton, G. W. and Johnston, R. D., (1986). Synthesis and Evaluation of Alternative Control Schemes for a Chalcopyrite Flotation Circuit. Int. J. Miner. Process., 16:29-42.

Austin, L. G., (1972). A Review Introduction to the Mathematical Descritption of Grinding as a Rate Process. Powder Technology, Vol 5, pp 1-17.

Austin, L. G. and Klimpel, R. R., (1984a). Modelling for Scale Up of Tumbling Ball Mills. Control '84, J. A. Herbst, editor. SME/AIME Publication, 1984, pp 167-184.

Austin, L. G., Klimpel, R. R. and Luckie, P. T., (1984b). Process Engineering of Size Reduction: Ball Milling. SME/AIME Publication, New York, 1984, 560 pp.

Bascur, O. A., (1982). Modelling and Computer Control of a Flotation Cell, Ph.D. Dissertation, University of Utah, Salt Lake City, Utah, (University Microfilms International, Ann Arbor), December 1982.

Bascur, O. A. and Herbst, J. A., (1985). Dynamic Simulators for Training Personnel in the Control of Grinding/ Flotation Systems. IFAC Automation for Mineral Resource Development. July 1985, pp 315-324.

Bekey, G. A., (1985). Simulability of Dynamical Systems. Trans. of The Society for Computer Simulation, Vol 2, No. 1, pp 57-71.

BMDP Statistical Software (1985). W. J. Dixon, Editor, University of California Press, Berkeley.

Bolles, J. L., Broderick, A. J. and Wampler, Jr., H. R., (1977). Morenci Concentrator Process Control System. Trans. SME/AIME, Vol. 262, Dec 1977, pp 310-319.

Box, M. J., Davies, D. and Swann, W. H., (1969). Non-Linear

Optimization Techniques, Monograph No. 5, Mathematical and Statistical Techniques for Industry. Published for Imperial Chemical Industries Ltd. by Oliver and Boyd, Edinburgh, UK. 60 pp.

Bradburn, R. G., Flintoff, B. C. and Walker, R. A., (1977). Practical Approach to Digital Control of a Grinding Circuit at Brenda Mines Ltd. SME AIME Transactions - Vol 262, pp 140-147. June 1977.

Brittan, M. I. and de Lange, W. F., (1975). Improving Gold Recoveries at the Western Holdings Concentrator with the Aid of Mathematical Modelling. Trans. Soc. of Mining Engineers, Vol 258, June 1975, pp 123-127.

de Jong, P. J. and de Wijn, W. J., (1984). Model Building and Simulation for Process Operator Training. Journal A, Vol 25, No. 1, pp 3-7.

Draper, N., Dredge, K. H. and Lynch, A. J., (1969). Operating Behaviour of an Automatic Control System for a Mineral Grinding Circuit. Proc. 9th Commonwealth Mining and Metallurgical Congress, Paper 22.

Finch, J. A. and Matwijenko, O., (1977). Individual Mineral Behaviour in a Closed Grinding Circuit. CIM Bulletin, November 1977, pp 164-172.

Finlayson, R. M. and Hulbert, D. G., (1980). The Simulation of the Behaviour of Individual Minerals in a Closed Grinding Circuit. IFAC Mining, Mineral and Metal Processing, Montreal, 1980, pp 323-332.

Flintoff, B. C., Plitt, L. R. and Turak, A. A., (1985). Cyclone Modelling: A Review of Present Technology. Paper presented at the 87th Ann. Gen. Meeting of CIM, Vancouver, B. C., April 1985.

Flintoff, B. C., Wong, W. Y. and Wood, R. K., (1985). Dynamic Simulation of Grinding Circuit Control. ISA Transactions, Vol 25, No. 1.

Ford, M. A. and King, R. P., (1984). The Simulation of Ore Dressing Plants. Int. J. Miner. Process., 12:285-304.

Franks, R. G. E., (1972). Modelling and Simulation in Chemical Engineering. John Wiley & Sons, New York, New York, 411 pp.

Franks, R. G. E., (1982). DYFLO Update: DYFLO2. Proc. Summer Conf. on Comp. Simulation. Denver, Colorado, pp 507-513.

Gault, G. A., Howarth, W. J., Lynch, A. J. and Whiten, W. J., (1979). Automatic Control of Grinding Circuits. World Mining. November 1979, pp 59-63.

Gossman, G. I., Hulbert, D. G., Koudstaal, J., Braae, M. and Buncombe, A., (1982). The Application of Computer Control to a Gold-Milling Circuit at East Driefontein Gold Mine. Proc. 12th CMMI Congress. H. W. Glen (editor). Johannesburg, South Africa, pp 817-825.

Govind, R. and Powers, G. J., (1982). Control System Synthesis Strategies. AIChE Journal, Vol 28, No 1, pp 60-73.

Harris, C. A. and Meech, J. A., (1985). Fuzzy Logic: A Potential Control Technique for Mineral Processing. 87th Annual General Meeting of CIM, paper No. 47, Vancouver, B. C., April 21-25, 1985.

Herbst, J. A. and Mular, A. L., (1979). Modelling and Simulation of Mineral Processing Unit Operations. Computer Methods for the 80's In the Mineral Industry, A. Weiss, editor, SME/AIME, New York, pp 823-840.

Herbst, J. A. and Oblad, A. E., (1985). Modern Control Theory Applied to Crushing Part 1: Development of a Dynamic Model for a Cone Crusher and Optimal Estimation of Crusher Operating Variables. IFAC Automation for Mineral Resource Development, A. W. Norris and D. R. Turner, editors. IFAC Proceedings Series, No. 1, 1986.

Herbst, J. A. and Rajamani, K., (1979). Control of Grinding Circuits. Computer Methods for the 80's In the Mineral Industry, A. Weiss, editor, SME/AIME, New York, pp 770-786.

Heyt, J. W. and Fairchild, B. T., (1982). Process Dynamic Simulators - A Potential User's Perspective. Proc. Summer Conf. Comp. Simul., Denver 1982, pp 548-553.

Kelly, E. G. and Spottiswood, D. J., (1982). Introduction to Mineral Processing. John Wiley & Sons, New York, New York, 480 pp.

Kelsall, D. F., Stewart, P. S. B. and Weller, K. R., (1978). Continuous Grinding in a Small Wet Ball Mill. Part V: A Study of the Influence of Media Shape. Powder Technology, Vol. 8, 1973, pp 77-83.

King, R. P., (1975). Simulation of Flotation Plants. Trans. SME/AIME, Vol. 258, Dec 1975, pp 286-301.

Laguitton, D., Pilgrim, R. F. and Seguin, F. X., (1983). Steady State Simulation of Comminution Circuits. Proc. Summer Conf. Comp. Simul., Vancouver 1983, pp 811-816.

Laguitton, D., ed. (1984). SPOC Manuals, CANMET Report SP85-1.

Lynch, A. J. and Rao, T. C., (1975). Modelling and Scale Up of Hydrocyclone Classifiers. 11th Int. Min. Proc.

Congress. Cagliari, Italy, April 1975, pp 245-269.

Lynch, A. J., (1977) Mineral Crushing and Grinding Circuits - Their Simulation, Optimisation, Design and Control. Developments in Mineral Processing 1, D. W. Furstenau, advisory editor, Elsevier Scientific Publishing Company, New York, 1977, 326 pp.

Lynch, A. J., (1984). Computers in Mineral Processing - the first twenty five years. 18th Int. Symp. - Application of Computers and Mathematics in the Mineral Industries. London England, March 1984, pp 1-10.

Manning, K. L., (1978). Computer Control of Grinding at Climax. Mining Congress Journal, November 1978, pp 47-50.

McDougall, G. B., (1987). Dynamic Simulation of the Control Behaviour of Mineral Grinding Circuits. MSc Thesis, University of Alberta, Spring 1987.

Niemi, A. J., (1983). Dynamic Models of Ore Dressing Processes, Their Steady State Counterparts and Use for Control. IFAC Automation in Mining, Mineral and Metal Processing, Helsinki Finland, 1983, pp 15-24.

Perkins, T. E. and Marnewecke, L., (1978). Automatic Grind Control at AMAX Lead Co.. Mining Engineering, February 1978, pp 166-170.

Plitt, L. R., (1971). The Analysis of Solid-Solid Separation in Classifiers. CIM Bulletin, April 1971, pp 42-47.

Plitt, L. R., (1976). A Mathematical Model of the Hydrocyclone Classifier. CIM Bulletin, Vol 69, pp 114-123.

Pritsker, A. A. B., (1984). Introduction to Simulation and SLAM II, 2nd Edition. John Wiley & Sons, N.Y., N.Y., 610 pp.

Rajamani, K., (1979). Optimal Control of Closed Circuit Ball Mill Grinding. Ph.D. Dissertation, University of Utah, Salt Lake City, Utah, (University Microfilms International, Ann Arbor), June 1979.

Ramirez-Castro, J. and Finch, J. A., (1980). Simulation of a Grinding Circuit Change to Reduce Lead Sliming. CIM Bulletin, April 1980, pp 132-137.

Rau, E. L. and Craig, R., (1982). Types and Characteristics of Common Grinding Circuit Flowsheets. Design and Installation of Comminution Circuits. SME Publication, Mular and Jergensen editors, pp 141-149.

Richardson, J. M., (1983). The Application of General Purpose Process Simulators in the Mining Industry. Proc. Summer Conf. Comp. Simul., Vancouver, B.C., 1983.

Sastry, K. V. S. and Adel, G. T., (1984). A Survey of Computer Simulation Software for Mineral Processing Systems. Control '84, J. A. Herbst, editor. SME/AIME Publication, 1984, pp 121-130.

Smith, C. A. and Corripio, A. B., (1985). Principles and Practice of Automatic Process Control. John Wiley & Sons, New York, New York, 609 pp.

Smith, H. W. and Guerin, D., (1980). Dynamic Modelling, Simulation and Control of a Grinding Circuit. CIM Bulletin, Sept 1980, pp 133-138.

Stephanopoulos, G., (1984). Chemical Process Control - An Introduction to Theory and Practice. Prentice Hall, Inc., New Jersey, USA, 696 pp.

Stewart, P. S. B., (1970). The Control of Wet Grinding Circuits. Australian Chemical Processing and Engineering. pp 22-33, June 1970.

Stewart, P. S. B. and Weller, K. R., (1977). Hold-up and Residence Time Characteristics of Full Scale Grinding Circuits. Paper presented at 3rd Iranian Congress of Chemical Engineering, 1977.

Spedden, H. R., Harper, R. W. and Herbst, J. A., (1970). Review of Mineral Processing, Mining Congress Journal, February 1970, pp 42-48.

The Simulation Councils Inc., (1967). The SCi Continuous System Simulation Language. Simulation, Vol 9, No 1. pp 281-304.

Tucker, R. J. and Meech, J. A., (1985). Modelling of Ore Crushing Plants for Teaching Purposes and Operator Training. CIM Proc. 17th Annual Meeting of the Canadian Mineral Processors. Ottawa, January 1985, pp 571-590.

Ulsoy, A. G. and Sastry, K. V. S., (1981). Principal Developments in the Automatic Control of Mineral Processing Systems. CIM Bulletin, December 1981, pp 43-53.

Whiten, W. J., (1972). The Simulation of Crushing Plants with Models Developed using Multiple Spline Regression. Journal of the South African Institute of Mining and Metallurgy. May 1972, pp 257-264.

Whiten, W. J., (1984). Models and Control Techniques for Crushing Plants. Control '84, J. A. Herbst, editor. SME/AIME Publication, 1984, pp 217-224.

Wills, B. A., (1985). Mineral Processing Technology, 3rd Edition. Pergamon Press, Oxford, U. K., 630 pp.

Winslow, R. L. and White, J. W., (1974). SIMUMILL - A User-Oriented Simulation System for Control Strategy Evaluation in Industrial Grinding Circuits. 12th Int. Symp. on the Application of Computers and Operations Research in the Mineral Industry, T. B. Johnson and D. W. Gentry, editors, pp B31-B47.

Wood, R. K. and Flintoff, B. C., (1982). Dynamic Process Simulation: CSSL Versus Process Simulation Program. Proc. Sum. Conf. Comp. Simul., Denver, 1982, pp 492-497.

```
#$Run PC.DRIVE.O+PC.CONFIG.O+PC.INPUTS.O+PC.MODELS.O+PC.DISTRB.O+PC.DYFLO.O+PC.CONTRL.O+PC.INIT.O 5='SOURCE' T=20S
#22:29:15

   This simulation package requires data to initialize
   the system.  Indicate which option is required for
   this particular run:

     1) use an established data file requiring no changes
     2) use an established data file with the option
        to make changes to some of the data
     3) interactively set up a new data file

   Enter the required option, press the ENTER key

2

   Enter the data file name - 12 characters or less
   2 for the drive label and colon,
   6 or less for the file name, and
   4 for the extension, e.g. .DAT

the.ssdata

   Indicate whether the data file contains just
   the minimum data required to start the simulator
   (OPTION 1), or the computer generated data file
   containing the complete steady state data
   (OPTION 2).

   ENTER 1 OR 2
2

Which changes in the data are required?

   1. Number of Size Classes
   2. Size Class Information
   3. Solids and Water Densities
   4. Mill Breakage and Selection Functions
   5. Mill Volume and Mixer Volumes
   6. Hydrocyclone Parameters
   7. Change Pump Model Only, and/or Kc on variable speed pump
   8. Sump/Pump Parameters
   9. STREAM 1 Size Distribution
  10. STREAM 1 Macro Properties
  11. RETURN

Enter one option at a time, the program will loop
around to allow you to make as many changes as
required.  Enter 11 to return to the main program
7
```

```
This option can only be used if the necessary
model parameters are included in the data file

If you are not sure, either check the data file,
lines 182-186, or go back to the menu, choose
option 8, load in the default values then return
here using option 7.
Currently the variable speed pump model is in effect
Enter the new option: 0-3


  0. Qin = Qout
  1. First Order Model - dh(t)/dt = Qin - k h**1/2
  2. Fixed Speed Pump
  3. Variable Speed Pump
3

Do you wish to set the proportional gain on the sump level controller
Answer Y(y) or N(n)
y

Enter the proportional gain Kc, values between 0.25 and 2.0 are reasonable
0.4

   Kc =   0.40
   Is this correct? - Ans Y(y) or N(n)
y

Anymore changes? - Ans Y(y) or N(n)
n

Input the simulation title

You are allowed up to 3 lines, 72 characters per
line. Just hit the return to input blank lines

This is a demonstration of the interactive nature of the simulator.
This is the PC version being run on the University of Alberta mainframe.
Open-loop simulation, 20 minutes, hardness disturbance.

The default initialization values are as follows:

  Simulation Initiation Time = 0.0 minutes
  Simulation Run Time = 100.0 minutes
  Incremental Step Size = 0.05 minutes
  Time Dependent Print Interval = 1.0 minutes
  Time of Disturbance Initiation = 2.0 minutes
  Open Loop Simulation (NO CONTROL)

Do you accept these values? - Y(y) or N(n)
n
```

```
Input one number from the menu at a time and the
program will allow you the option of looping back
to make further changes

Indicate which initialization value requires
changing.

   1. Simulation Initiation Time
   2. Simulation Run Time
   3. Incremental Step Size
   4. Time Dependent Print Interval
   5. Time of Disturbance Initiation
   6. Flag No Disturbance Initiation
   7. Choice of Control Algorithm

2

    The current simulation time is 100.00 minutes

    Input the required simulation time
20.

Do you require any further changes in the
initialization values? - Ans Y(y) or N(n)
n


    Are these the required initialization values?
    Simulation Initiation Time =  0.0 minutes
    Simulation Run Time =  20.00 minutes
    Incremental Step Size = 0.050 minutes
    Time Dep. Print Interval =  1.00 minutes
    Time of Disturbance Initiation =  2.00 minutes
    Disturbance Flag = ON
    Type of Simulation = OPEN LOOP

    Are these values correct? If you answer in the affirmative then
    the simulation will begin
Y

Enter the datafile name for the time dependent
circuit variables.  If this is a previously used
file all previous data will be destroyed as
its overwritten - 6 chars. or less in filename
plus 4 for the extension, e.g. .DAT.

A:EXAMPLE
  Enter the file status - NEW or OLD
NEW
```

```
Enter the type of disturbance required

   0. No Disturbance
   1. Change in feedstream components
   2. Change in feedstream mass flowrate
   3. Change in feedstream size distribution
   4. Change in extraneous water to the sump

Enter an integer value between 1 and 4

Option 4 not valid for the open circuit configuration

1

Enter the required type of component change

   1. Step change in feed components, ie model
                    a step change in hardness
   2. Single ramp change in feed components,
        ie model a gradual change in hardness
   3. Multiple ramp changes in feed components
   4. Square wave changes in feed components

Enter the number of the option

1

   Currently the feed stream is 100.00% component #1

   Enter the required percentage of component #1 at the end of
   the step function - 0.0 <= % <= 100.0
0.0

  Do you want to save the mill breakage and selection
  elements, and the stream synopsis for this
  simulation run?  - Enter Y(y) or N(n)
Y
Enter the data file name - 12 characters or less
B.EXAMPLE
  Enter the file status - NEW or OLD
NEW

  Do you want to save the steady state data
  to a permanent file for future reference?

  Answer Y(y) or N(n)
Y

  Enter the data file name - 12 characters of less
C.EXAMPLE
  Enter the file status - NEW or OLD
NEW
```

Simulation Title: This is a demonstration of the interactive nature of the simulator.
This is the PC version being run on the University of Alberta mainframe.
Open-loop simulation, 20 minutes, hardness disturbance.

SIMULATION TIME IN MINUTES	FRESH ORE FEEDRATE (tph)	CYCLONE FEED DENSITY (%)	PUMP SUMP HEIGHT (M)	WATER MASS FLOW RATE TO SUMP (tph)	BALL MILL DISCHARGE -150 MIC FREQUENCY	BALL MILL DISCHARGE SOLIDS MASS FLOW RATE	RECIRC. LOAD (%)	PRODUCT STREAM -150 MIC FREQUENCY	PRODUCT STREAM SOLIDS MASS FLOW RATE
0.0	6.563E+01	6.723E+01	1.006E+00	4.802E+01	6.604E+01	1.219E+02	1.857E+02	8.100E+01	6.559E+01
1.000E+00	6.563E+01	6.723E+01	1.007E+00	4.802E+01	6.604E+01	1.218E+02	1.856E+02	8.100E+01	6.557E+01
2.000E+00	6.563E+01	6.723E+01	1.007E+00	4.802E+01	6.604E+01	1.218E+02	1.856E+02	8.100E+01	6.557E+01
3.000E+00	6.563E+01	6.723E+01	1.007E+00	4.802E+01	6.604E+01	1.218E+02	1.856E+02	8.100E+01	6.557E+01
4.000E+00	6.563E+01	6.723E+01	1.007E+00	4.802E+01	6.604E+01	1.218E+02	1.857E+02	8.100E+01	6.558E+01
5.000E+00	6.563E+01	6.723E+01	1.007E+00	4.802E+01	6.604E+01	1.218E+02	1.856E+02	8.101E+01	6.556E+01
6.000E+00	6.563E+01	6.723E+01	1.007E+00	4.802E+01	6.603E+01	1.219E+02	1.857E+02	8.102E+01	6.548E+01
7.000E+00	6.563E+01	6.723E+01	1.007E+00	4.802E+01	6.599E+01	1.219E+02	1.858E+02	8.104E+01	6.539E+01
8.000E+00	6.563E+01	6.723E+01	1.008E+00	4.802E+01	6.590E+01	1.220E+02	1.859E+02	8.106E+01	6.528E+01
9.000E+00	6.563E+01	6.723E+01	1.008E+00	4.802E+01	6.575E+01	1.221E+02	1.860E+02	8.107E+01	6.516E+01
1.000E+01	6.563E+01	6.724E+01	1.008E+00	4.802E+01	6.555E+01	1.222E+02	1.862E+02	8.108E+01	6.502E+01
1.100E+01	6.563E+01	6.725E+01	1.009E+00	4.802E+01	6.530E+01	1.224E+02	1.864E+02	8.107E+01	6.486E+01
1.200E+01	6.563E+01	6.725E+01	1.010E+00	4.802E+01	6.500E+01	1.225E+02	1.867E+02	8.106E+01	6.469E+01
1.300E+01	6.563E+01	6.727E+01	1.011E+00	4.802E+01	6.467E+01	1.227E+02	1.870E+02	8.102E+01	6.451E+01
1.400E+01	6.563E+01	6.728E+01	1.012E+00	4.802E+01	6.430E+01	1.230E+02	1.874E+02	8.098E+01	6.432E+01
1.500E+01	6.563E+01	6.730E+01	1.013E+00	4.802E+01	6.390E+01	1.232E+02	1.878E+02	8.091E+01	6.413E+01
1.600E+01	6.563E+01	6.732E+01	1.015E+00	4.802E+01	6.349E+01	1.235E+02	1.882E+02	8.083E+01	6.394E+01
1.700E+01	6.563E+01	6.734E+01	1.016E+00	4.802E+01	6.305E+01	1.238E+02	1.886E+02	8.072E+01	6.375E+01
1.800E+01	6.563E+01	6.737E+01	1.018E+00	4.802E+01	6.259E+01	1.241E+02	1.891E+02	8.059E+01	6.357E+01
1.900E+01	6.563E+01	6.739E+01	1.020E+00	4.802E+01	6.214E+01	1.245E+02	1.896E+02	8.045E+01	6.341E+01
2.000E+01	6.563E+01	6.742E+01	1.022E+00	4.802E+01	6.169E+01	1.248E+02	1.902E+02	8.029E+01	6.326E+01

DYNAMIC SIMULATION OF A GRINDING CIRCUIT

Simulation Title: This is a demonstration of the interactive nature of the simulator.
This is the PC version being run on the University of Alberta mainframe.
Open-loop simulation, 20 minutes, hardness disturbance.

ROD MILL SPECIFICATIONS FOLLOW...

PULP HOLDUP VOLUME(M**3)= 3.9

FOR COMPONENT #1

LOWER BOUND ON SIZE	FIRST ORDER SELECTION CONSTANT	FIRST COLUMN OF BREAKAGE MATRIX
4757.	22.950	.0
3364.	4.860	.2090
2378.	1.028	.1650
1682.	0.218	.1310
1189.	0.046	.1030
841.	0.021	.0820
595.	0.021	.0650
420.	0.021	.0510
297.	0.021	.0410
210.	0.021	.0320
149.	0.021	.0250
105.	0.021	.0200
74.	0.021	.0160
53.	0.021	.0130
37.	0.021	.0100
10.	0.0	.0

FOR COMPONENT #2

LOWER BOUND ON SIZE	FIRST ORDER SELECTION CONSTANT	FIRST COLUMN OF BREAKAGE MATRIX
4757.	11.475	.0
3364.	2.430	.2090
2378.	0.514	.1650
1682.	0.109	.1310
1189.	0.023	.1030
841.	0.011	.0820
595.	0.011	.0650
420.	0.011	.0510
297.	0.011	.0410
210.	0.011	.0320
149.	0.011	.0250
105.	0.011	.0200
74.	0.011	.0160
53.	0.011	.0130
37.	0.011	.0100
10.	0.0	.0

BALL MILL SPECIFICATIONS FOLLOW...

PULP HOLDUP VOLUME(M**3)= 8.5

FOR COMPONENT #1

LOWER BOUND ON SIZE	FIRST ORDER SELECTION CONSTANT	FIRST COLUMN OF BREAKAGE MATRIX
4757.	33.304	.0
3364.	17.739	.2020
2378.	9.441	.1610
1682.	5.032	.1290
1189.	2.679	.1030
841.	1.427	.0820
595.	0.761	.0650
420.	0.404	.0520
297.	0.215	.0420
210.	0.114	.0330
149.	0.061	.0270
105.	0.033	.0210
74.	0.017	.0170
53.	0.009	.0130
37.	0.003	.0110
10.	0.0	.0

FOR COMPONENT #2

LOWER BOUND ON SIZE	FIRST ORDER SELECTION CONSTANT	FIRST COLUMN OF BREAKAGE MATRIX
4757.	16.652	.0
3364.	8.869	.2020
2378.	4.720	.1610
1682.	2.516	.1290
1189.	1.339	.1030
841.	0.714	.0820
595.	0.380	.0650
420.	0.202	.0520
297.	0.107	.0420
210.	0.057	.0330
149.	0.030	.0270
105.	0.016	.0210
74.	0.009	.0170
53.	0.005	.0130
37.	0.001	.0110
10.	0.0	.0

Simulation Title: This is a demonstration of the interactive nature of the simulator
This is the PC version being run on the University of Alberta mainframe
Open-loop simulation, 20 minutes, hardness disturbance.

STREAM ATTRIBUTES AT TIME 20.05

SIZE	STRM 1	STRM 2	STRM 3	STRM 4	STRM 5	STRM 6	STRM 7	STRM 8	STRM 9	STRM 10	STRM 11	STRM 12	STRM 13	STRM 14	STRM 15
FOR COMPONENT #1															
4757.	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000	0.000	0.000	0.0	0.000	0.000	0.000	0.0
3364.	0.0	0.0	0.0	0.0	0.000	0.000	0.0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
2378.	0.0	0.0	0.0	0.0	0.000	0.000	0.0	0.000	0.001	0.001	0.000	0.001	0.000	0.000	0.000
1682.	0.0	0.0	0.0	0.0	0.010	0.015	0.0	0.007	0.028	0.031	0.000	0.046	0.002	0.002	0.000
1189.	0.0	0.0	0.0	0.0	0.368	0.454	0.0	0.185	0.331	0.354	0.001	0.531	0.034	0.043	0.001
841.	0.0	0.0	0.0	0.0	0.583	0.697	0.0	0.330	0.529	0.560	0.013	0.835	0.111	0.135	0.014
595.	0.0	0.0	0.0	0.0	0.518	0.617	0.0	0.442	0.678	0.715	0.067	1.041	0.288	0.347	0.072
420.	0.0	0.0	0.0	0.0	0.446	0.529	0.0	0.812	1.138	1.189	0.295	1.637	0.817	0.953	0.312
297.	0.0	0.0	0.0	0.0	0.399	0.471	0.0	1.797	2.263	2.335	1.121	2.945	2.217	2.480	1.167
210.	0.0	0.0	0.0	0.0	0.356	0.419	0.0	3.450	4.021	4.109	3.086	4.623	4.644	5.024	3.170
149.	0.0	0.0	0.0	0.0	0.317	0.373	0.0	4.977	5.568	5.658	5.731	5.622	6.950	7.378	5.845
105.	0.0	0.0	0.0	0.0	0.281	0.330	0.0	5.502	6.042	6.125	7.623	5.373	7.796	8.202	7.750
74.	0.0	0.0	0.0	0.0	0.250	0.292	0.0	5.176	5.640	5.712	8.173	4.477	7.372	7.726	8.298
53.	0.0	0.0	0.0	0.0	0.230	0.270	0.0	4.420	4.814	4.874	7.642	3.485	6.290	6.588	7.759
37.	0.0	0.0	0.0	0.0	0.181	0.211	0.0	3.519	3.822	3.869	6.492	2.552	5.015	5.247	6.588
10.	0.0	0.0	0.0	0.0	1.173	1.351	0.0	13.110	14.382	14.577	27.336	8.175	18.331	19.248	27.796
FOR COMPONENT #2															
4757.	61.000	61.000	0.0	61.000	0.872	0.872	0.0	0.303	0.304	0.304	0.0	0.457	0.005	0.005	0.0
3364.	8.000	8.000	0.0	8.000	1.318	1.318	0.0	0.464	0.465	0.465	0.000	0.699	0.015	0.015	0.000
2378.	6.000	6.000	0.0	6.000	4.877	4.876	0.0	1.744	1.747	1.747	0.000	2.624	0.097	0.097	0.000
1682.	5.240	5.240	0.0	5.240	11.647	11.617	0.0	4.286	4.225	4.215	0.001	6.330	0.437	0.432	0.001
1189.	2.840	2.840	0.0	2.840	13.149	13.028	0.0	5.136	4.939	4.908	0.016	7.363	1.013	0.988	0.016
841.	1.800	1.800	0.0	1.800	10.924	10.802	0.0	4.877	4.616	4.575	0.108	6.817	1.830	1.765	0.106
595.	1.600	1.600	0.0	1.600	8.929	8.827	0.0	4.906	4.537	4.479	0.422	6.516	3.007	2.853	0.412
420.	1.040	1.040	0.0	1.040	6.941	6.859	0.0	4.841	4.332	4.253	1.055	5.858	4.087	3.796	1.024
297.	1.090	1.090	0.0	1.090	5.839	5.769	0.0	4.786	4.171	4.076	1.957	5.140	4.701	4.290	1.888
210.	1.160	1.160	0.0	1.160	4.934	4.874	0.0	4.370	3.751	3.656	2.745	4.113	4.574	4.128	2.639
149.	1.210	1.210	0.0	1.210	4.201	4.149	0.0	3.780	3.225	3.140	3.180	3.119	4.018	3.606	3.053
105.	1.170	1.170	0.0	1.170	3.565	3.521	0.0	3.149	2.685	2.614	3.253	2.293	3.318	2.971	3.124
74.	1.130	1.130	0.0	1.130	3.050	3.012	0.0	2.596	2.219	2.161	3.092	1.694	2.670	2.390	2.972
53.	1.360	1.360	0.0	1.360	2.886	2.852	0.0	2.266	1.953	1.905	2.987	1.362	2.196	1.969	2.877
37.	0.520	0.520	0.0	0.520	1.809	1.785	0.0	1.563	1.331	1.295	2.174	0.855	1.628	1.456	2.088
10.	4.840	4.840	0.0	4.840	9.947	9.811	0.0	7.204	6.243	6.095	11.430	3.418	6.539	5.867	11.028
MF SOLIDS	65.63	65.63	0.0	65.63	65.63	65.63	0.0	190.44	189.21	189.16	63.21	125.95	125.24	124.84	63.25
MF WATER	0.0	0.0	11.58	11.58	11.58	11.58	48.02	91.67	91.37	91.39	59.59	31.79	32.09	32.07	59.57
VOL. FLOW	27.35	27.35	11.58	38.93	38.93	38.93	48.02	171.02	170.20	170.20	85.93	84.27	84.27	84.09	85.93
% SOLIDS	100.00	100.00	0.0	85.00	85.00	85.00	0.0	67.50	67.44	67.43	51.47	79.84	79.60	79.56	51.50